LAB-82-03

Permanent Counting Stations (PCS)

Microcomputer Programming Manual

S. Skelton
Research Engineer
Experimental Demonstration & Testing

Published by:
The Transportation Technology and Energy Division
Ontario Ministry of Transportation and Communications
Hon. James W. Snow, Minister
H.F. Gilbert, Deputy Minister

Published without prejudice
as to the application of the findings.
Crown copyright reserved; however, this
document may be reproduced for non-commercial
purposes with attribution to the Ministry.

For additional copies write:
The Editor
Research and Development Branch
Ministry of Transportation and Communications
1201 Wilson Avenue
Downsview, Ontario
M3M 1J8

September 1982

TABLE OF CONTENTS PAGE

1/ INTRODUCTION 1

2/ SYSTEM OVERVIEW 2
2.1/ General Description 2
2.2/ Road to Memory Lane Correlation 9
2.3/ Ode to Speed 9
2.4/ Lane Proportioning Constants. 12

3/ RAM PAGES 14
3.1/ Page ØØxy 14
3.2/ Page Ø1xy 14
3.3/ Page Ø2xy 19
3.4/ Page Ø3xy 19
3.5/ Page Ø4xy 19

4/ OPERATING PROCEDURES 27

5/ ROUTINE SUMMARY 30
5.1/ Lane Data Pointer Initialization 30
5.2/ Lane Scan Control 31
5.3/ 8-Bit Multiplication 32
5.4/ 8-Bit Division 33
5.5/ 16-Bit Number Comparison 34
5.6/ Lane Proportioning Control 35
5.7/ System Initialization 36
5.8/ Speed Output and Header Control 37
5.9/ Detector Input Scan Control 38
5.10/ Single Error Correction Routine 39
5.11/ Upstream Detector Servicing Routine. 40
5.12/ Downstream Detector Servicing Routine. 41
5.13/ Lane Data Handling Routine 42
5.14/ Average Running Total Routine 43
5.15/ Detector Failure Check Routine 44
5.16/ Hourly Data Transfer Routine 46
5.17/ Vehicle Speed and Length Classification 47

PAGE

6/ ROUTINE DESCRIPTION 48
6.1/ 'SETUP' 48
6.1.1/ Program Listing Reference. 48
6.1.2/ Program Explanation. 49
6.1.3/ Parameter Index. 50
6.1.4/ Flow Chart 51
6.2/ 'NEXTLN'. 52
6.2.1/ Program Listing Reference. 52
6.2.2/ Program Explanation. 53
6.2.3/ Parameter Index. 54
6.2.4/ Flow Chart 55
6.3/ 'MULT' 56
6.3.1/ Program Listing Reference. 56
6.3.2/ Program Explanation. 57
6.3.3/ Parameter Index. 58
6.3.4/ Flow Chart 59
6.4/ 'DIV' 61
6.4.1/ Program Listing Reference. 61
6.4.2/ Program Explanation. 62
6.4.3/ Parameter Index. 63
6.4.4/ Flow Chart 64
6.5/ 'CMP16' 66
6.5.1/ Program Listing Reference. 66
6.5.2/ Program Explanation. 67
6.5.3/ Parameter Index. 68
6.5.4/ Flow Chart 69
6.6/ 'PRPRT/PRPRT8'. 71
6.6.1/ Program Listing Reference. 71
6.6.2/ Program Explanation. 72
6.6.3/ Parameter Index. 75
6.6.4/ Flow Chart 76
6.7/ 'INIT' 78
6.7.1/ Program Listing Reference. 78
6.7.2/ Program Explanation. 79
6.7.3/ Parameter Index. 81
6.7.4/ Flow Chart 86

PAGE

6.8/ 'HEADER' 89
6.8.1/ Program Listing Reference. 89
6.8.2/ Program Explanation. 90
6.8.3/ Parameter Index. 92
6.8.4/ Flow Chart 94
6.9/ 'INPUT' 96
6.9.1/ Program Listing Reference. 96
6.9.2/ Program Explanation. 97
6.9.3/ Parameter Index. 98
6.9.4/ Flow Chart 99
6.10/ 'PCS2' 102
6.10.1/ Program Listing Reference 102
6.10.2/ Program Explanation 103
6.10.3/ Parameter Index 105
6.10.4/ Flow Chart. 107
6.11/ 'PCS3' 108
6.11.1/ Program Listing Reference 108
6.11.2/ Program Explanation 109
6.11.3/ Parameter Index 110
6.11.4/ Flow Chart. 112
6.12/ 'PCS3B' 114
6.12.1/ Program Listing Reference 114
6.12.2/ Program Explanation 115
6.12.3/ Parameter Index 116
6.12.4/ Flow Chart. 117
6.13/ 'PCS4' 118
6.13.1/ Program Listing Reference 118
6.13.2/ Program Explanation 119
6.13.3/ Parameter Index 120
6.13.4/ Flow Chart. 123
6.14/ 'PCS5' 126
6.14.1/ Program Listing Reference 126
6.14.2/ Program Explanation 127
6.14.3/ Parameter Index 128
6.14.4/ Flow Chart. 130

PAGE

6.15/ 'PCS6' 132
6.15.1/ Program Listing Reference 132
6.15.2/ Program Explanation 133
6.15.3/ Parameter Index 134
6.15.4/ Flow Chart. 137
6.16/ 'PCS7' 139
6.16.1/ Program Listing Reference 139
6.16.2/ Program Explanation 140
6.16.3/ Parameter Index 142
6.16.4/ Flow Chart. 145
6.17/ 'PCS8' 146
6.17.1/ Program Listing Reference 146
6.17.2/ Program Explanation 147
6.17.3/ Parameter Index 150
6.17.4/ Flow Chart. 153

7/ SYSTEM DEVELOPMENT 157

8/ ACKNOWLEDGEMENTS 158

9/ BIBLIOGRAPHY 159

Appendix A/ Equate File 161
Program Listing 162
Cross-Reference Table 193

1/ INTRODUCTION

The purpose of this programming manual is to offer enough background information to a user to afford him or her a basic understanding of the system. The text does not deal with its respective subject in great detail, but, instead, tries to deal with the salient points. This turns out to be both an advantage and disadvantage: the brevity limits the quantity of text to be digested, but if the assumption that the reader has programming tendencies is in error, the reader may find portions difficult to follow because of the lack of development.

The manual deals with the completion of Phase II software. This version features up to 16 lanes of monitoring vehicle counts, long vehicles and speed. It does so by the manipulation of 26 routines. Of these, 10 were written directly for PCS, and the other 16 were taken from Caltrans District 11 software supplied for ramp metering. Modification to the Caltrans software ranges from minor to extensive, sometimes leaving little resemblance to the original form.

2/ SYSTEM OVERVIEW

The software for PCS was developed in a modular form. Each module treats a specific area of interest. Besides the tendency of single concept per module, there is also a distinction made for traffic direction. The routines carrying an 'A' suffix are Direction 2 routines, which are like Direction 1 with their references to memory offset. The exception to this convention is in 'PCS7' and 'PCS7A' where all volumes and all lengths, respectively, are dealt with.

2.1/ General Description

The routine layout, for the 26 routines, can be seen in Figure 2.1. The initialization routine is not called in the normal run of things; it must be invoked separately. Once the program is initialized and running, the amount of memory required depends on the size and configuration of the station. But generally, RAM Pages ØØ through Ø4 are represented in Figures 2.2 through 2.6.

Page Ø holds the temporary storage area for PCS parameters in the general storage area, as well as some specific transient lane data. It also has the communications area, lane configuration and timing data.

Page 1 is the main storage area for lane data. It holds the various forms of the occupancy and volume. It also stores the long vehicle counts calculated by the 'PCS8' series program. All of the information is organized on a per-lane basis, first in Direction 1, then Direction 2.

Page 2 contains all the information for the calculation and classification of speed. There are the transition timing bytes for both directions, plus the roll over flag to ensure timing integrity. Also, there are static values for loop spacing, dimensioning and the threshold values used as standard to classify speeds. To that end, there is also the speed groups storage area which maintains a 3-byte count of each of the 12 speed categories, from >130 km/h to <30 km/h.

Page 3 is simply an extension of the previous page, since it deals with the daily storage of speed data. Whenever the 'HEADER' routine is

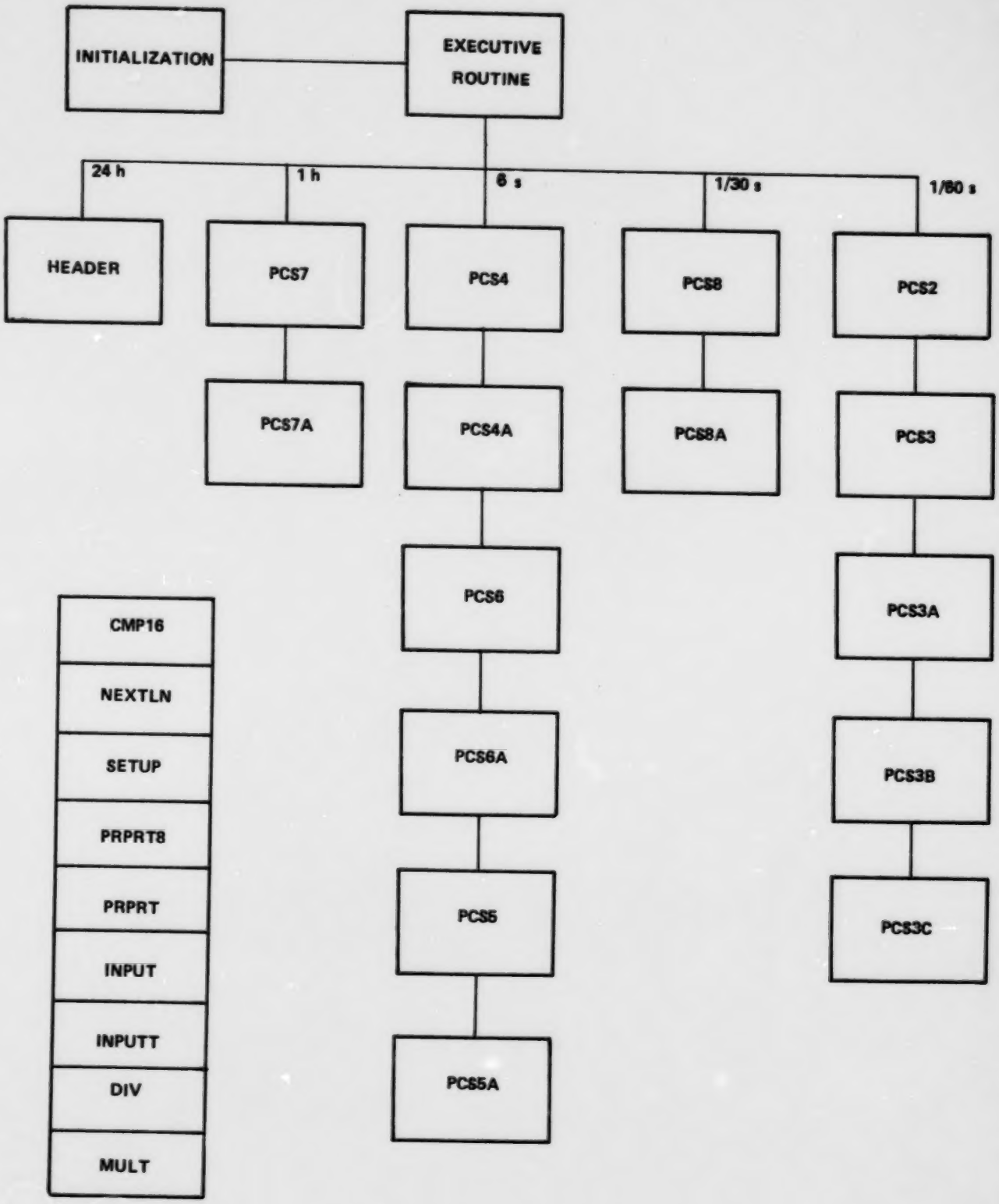


Figure 2.1/ Routine Layout

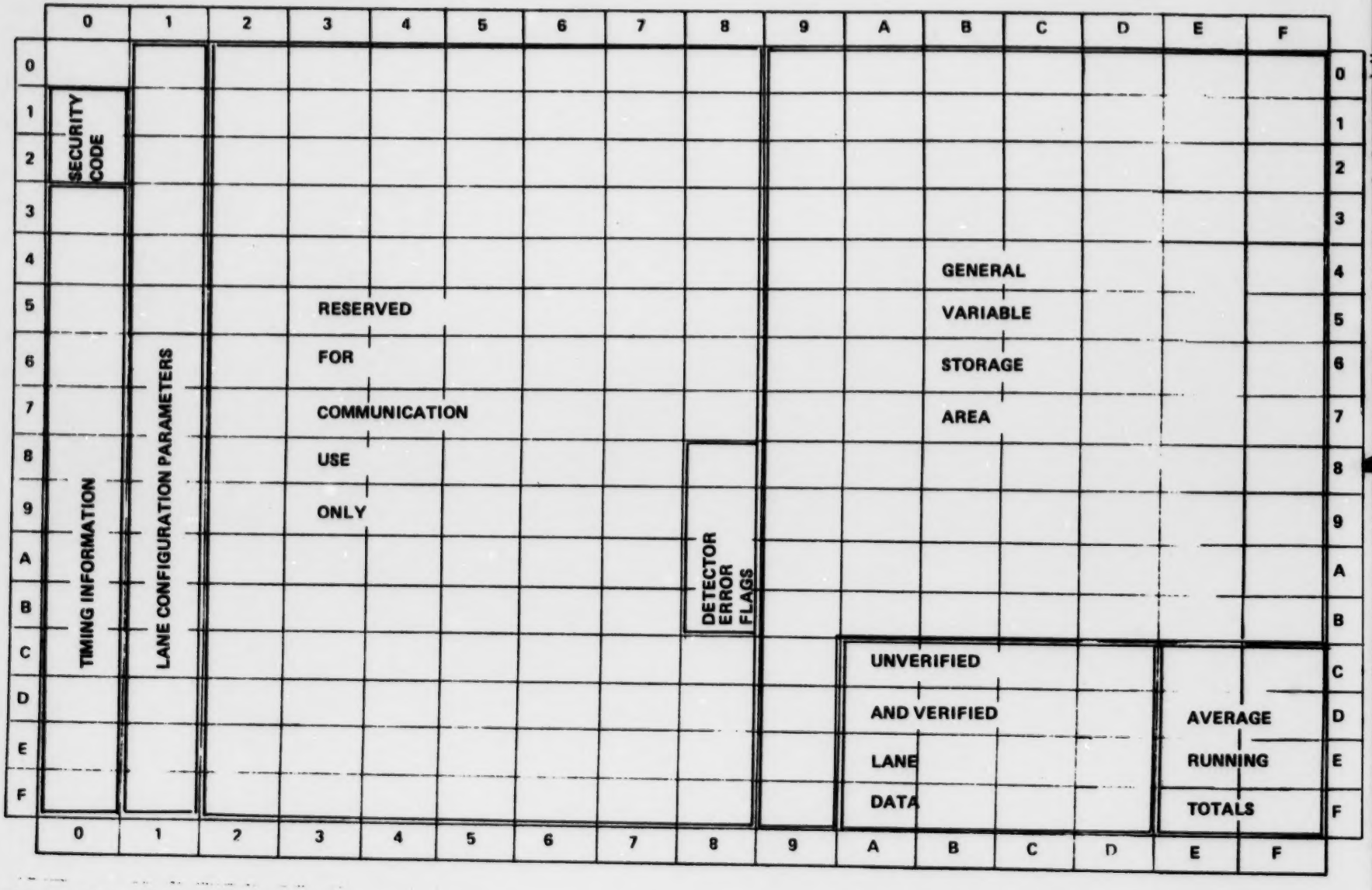


Figure 2.2/ Typical RAM Page ϕϕxy

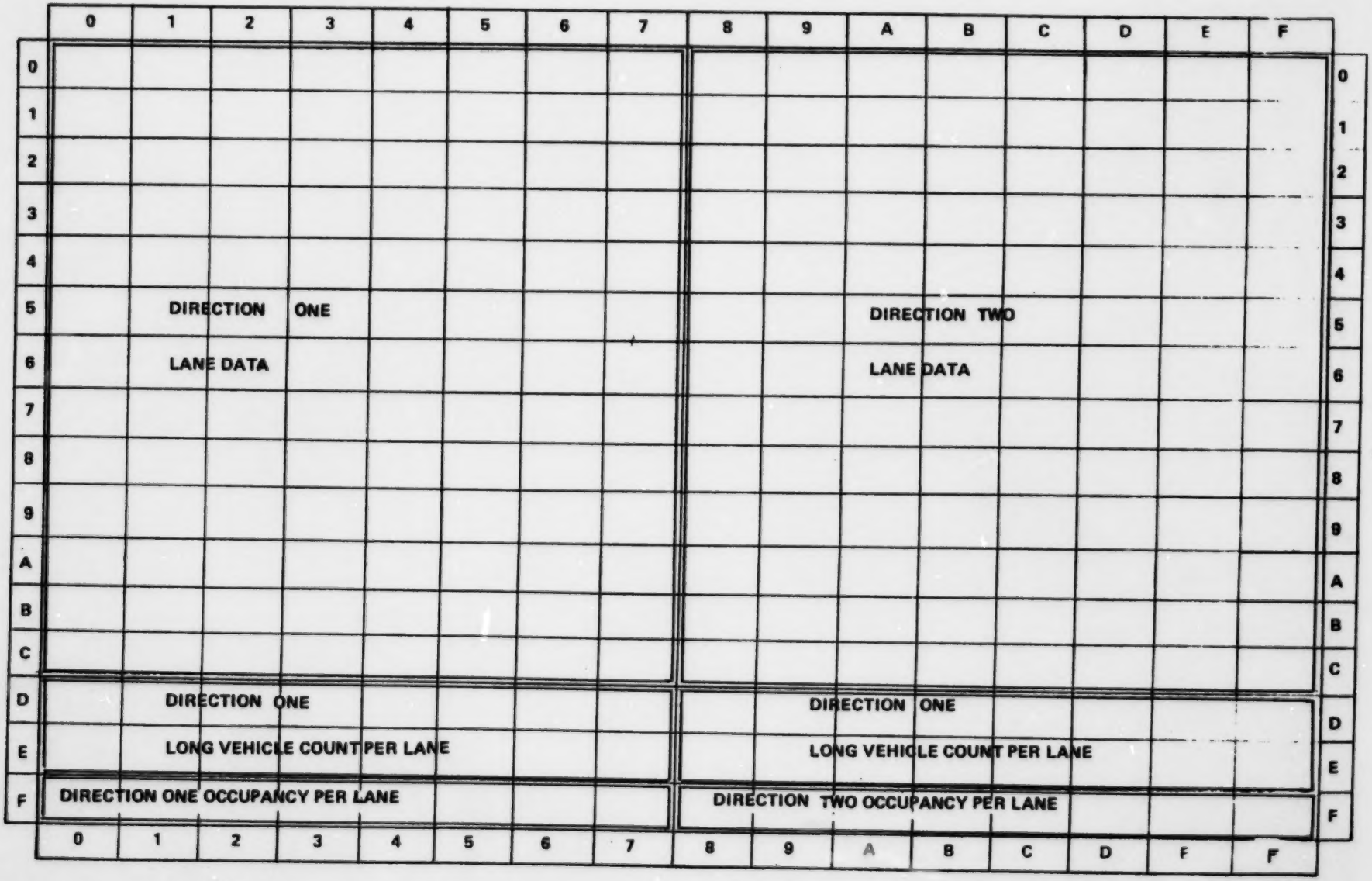


Figure 2.3/ Typical RAM Page ϕ1xy

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0				DIRECTION ONE							MSB	ISB	LSB	> 130			0
1				LANE TRANSITION TIMING BLOCK													1
2				DIRECTION TWO													2
3				LANE TRANSITION TIMING BLOCK													3
4				DIRECTION ONE LOOP SPACINGS													4
5				DIRECTION TWO LOOP SPACINGS					SPEED GROUP THRESHOLD	VALUES		SPEED					5
6				DIRECTION ONE LOOP DIMENSIONS								GROUP					6
7				DIRECTION TWO LOOP DIMENSIONS								DATA					7
8				DIRECTION ONE													8
9				COUNTER ROLL OVER FLAG													9
A				DIRECTION TWO													A
B				COUNTER ROLL OVER FLAG										< 30			B
C																	C
D																	D
E																	E
F																	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Figure 2.4/ Typical RAM Page ϕ2xy

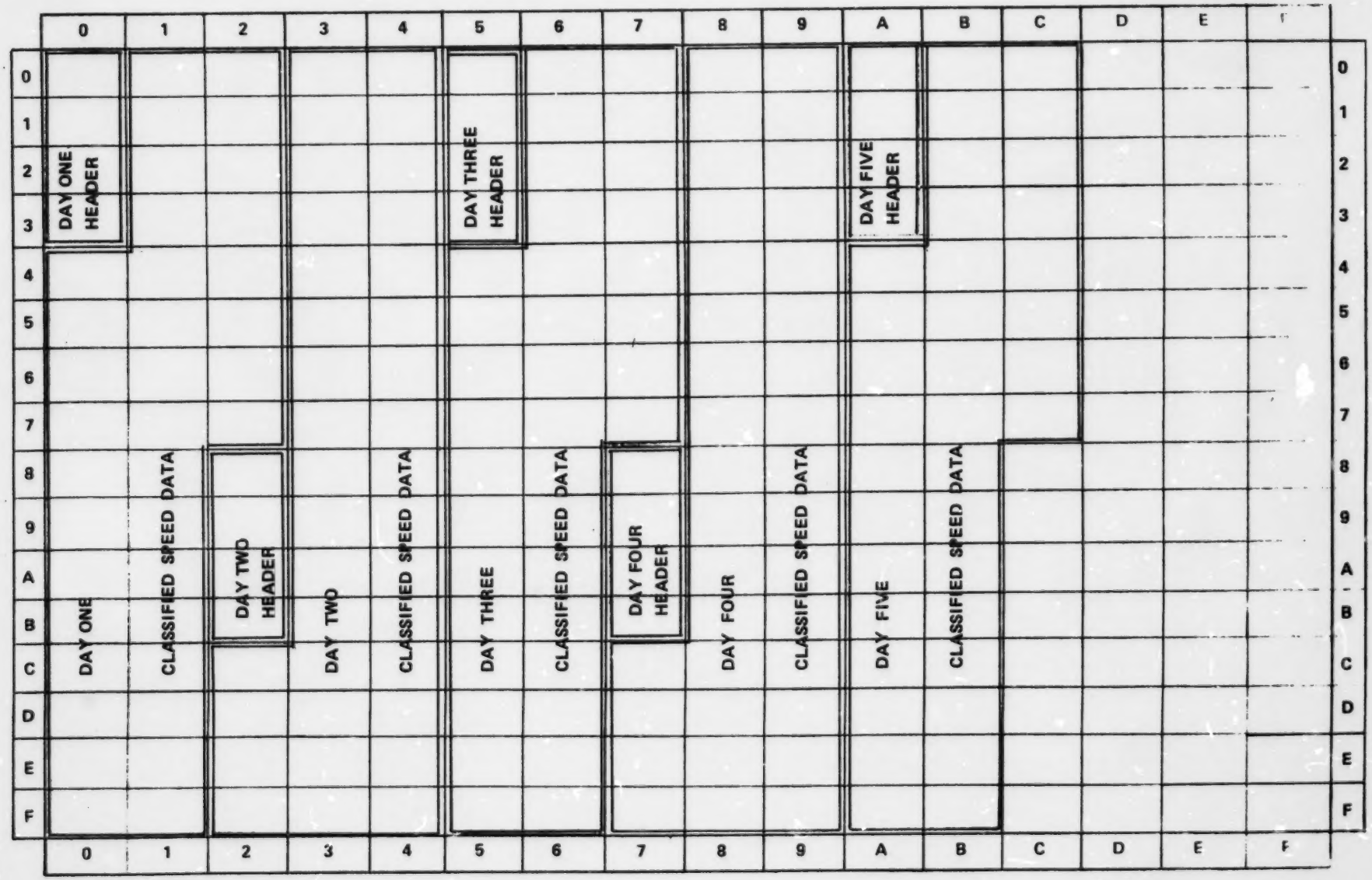


Figure 2.5/ Typical RAM Page ϕ3xy

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	DAY ONE VOLUME HEADER																0
1																	1
2																	2
3			DAY ONE														3
4			VOLUME														4
5			TRANSMIT														5
6			BUFFER														6
7																	7
8																	8
9																	9
A																	A
B																	B
C																	C
D																	D
E																	E
F																	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Figure 2.6/ Typical RAM Page ϕ4xy

invoked, once every 24 h, its first duty is to transfer grouped speed data from Page 2 into the appropriate area in the speed transmit buffer. Day 1 is initialized to Location Ø3ØØ in all cases. The days are laid out as seen in Figure 2.5, where the sixth day would overlap Day 1, and so on.

Page 4 is the start of the volume counts. Day 1 is always initialized to Ø4ØØ, while the position of the Day 2 header depends upon the size of the station and configuration chosen. This is also true for the long vehicle count buffer which follows, since it is identical in format to the volume buffer, but its origin is based on the size of the volume buffer.

2.2/ Road to Memory Lane Correlation

The system is designed to handle up to 16 lanes, resulting in the possibility of up to 32 loop detectors. The software relates the road configuration with the internal storage area. For a user, Page 1 needs, for the most part, no explanation, but the instantaneous values of Page Ø are not obvious.

Figure 2.7 demonstrates the relationship between the road and memory, which is useful to check the operation of the loop detector amplifiers and input software routines. Basically, the verified detector input data resides in Page Ø, Row E, Columns A, B, C and D. These addresses are updated every 1/60 of a second and give a BCD version of the binary lane designation.

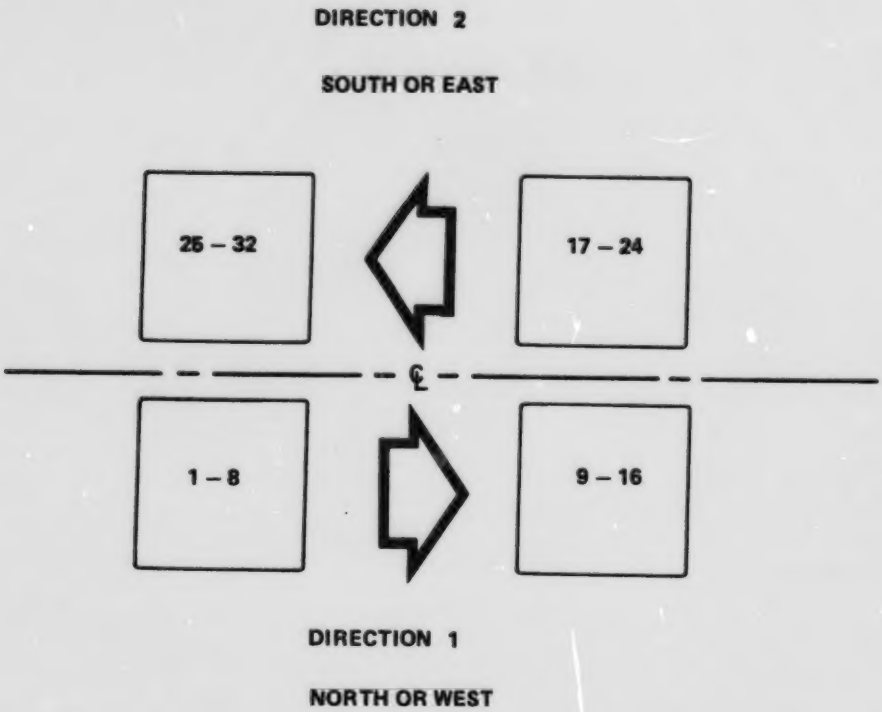
2.3/ Ode to Speed

A vehicle entering the test sections, besides being counted, is subjected to speed and length estimations which are determined by the following equations:

$$\text{Speed} = \frac{1}{2}\left[\frac{LS}{B1E} + \frac{LS}{B1S}\right] \qquad (1)$$

where: LS = loop spacing

B1E = vehicle end transition timer



INSTANTANEOUS DETECTOR STATUS

ϕϕxy	A	B	C	D	
	1 – 8	17 – 24	9 – 16	25 – 32	E

i.e., lane Direction 1 upstream located ϕϕAE

Figure 2.7/ Road to Memory Correlation

B1S = vehicle start transition timer

$$\text{Length} = \text{OCC}*\text{SPEED} - \text{LD} \qquad (2)$$

where: OCC = vehicle occupancy of upstream detectors
LD = loop dimensions

Therefore, from Equation 1, any speed can be calculated given the requisite parameters. The units of km/h are not used, since loop spacing is in feet and timing is in 1/60 of a second counts. As a result, a new set of speed standards was derived. These standards are known as the threshold values, and are a result of all linear dimensions times eight and time in 1/60 of a second counts. These threshold values, translated into hex, are stored on RAM Page 2. A full speed conversion table is given in Figure 2.8, while a detailed example of one follows.

Given: vehicle travelling at 100 km/h (62.2 mph)
Objective: to determine the speed threshold values for given speeds

From (1):

$$S = \frac{1}{2}\left[\frac{LS}{B1E} + \frac{LS}{B1S}\right]$$

Where: LS = 20 ft. x 8 = 160 ft.
Now: B1E = B1S = passage time in sample intervals

Therefore: $62.2 \text{ mph} = \frac{62.2 \times 5280}{3600} = 91.2 \text{ ft./s}$

Passage Time $= \frac{20 \text{ ft.}}{91.2} \; \frac{s}{\text{ft.}} = 219.2 \text{ ms}$

Since: 219.2 ms is required to cover the test section, and the sampling rate is 1/60 of a second, the total number of counts is:

$$\text{Total number of counts} = \frac{219.2}{16.7} = 13.2 \text{ counts}$$

Therefore, threshold value from (1):

$$S = \frac{1}{2}\left[\frac{160 \text{ ft.}}{13} + \frac{160 \text{ ft.}}{13}\right]$$

$$S = 12.30$$

Note: The loop spacing and dimension values both are represented in memory by a hex value which is eight times the linear value in feet. The default value of LS is equivalent to 26 ft., while LD is equivalent to 6 ft.

km/h	mph	ft./s	$\frac{\text{ft.x 8}}{\text{count}}$	SPD	REM	Speed Group
130	80.8	118.5	15.8	15	48	>130
120	74.6	109.4	14.6	14	36	120 - 130
110	68.4	100.3	13.4	13	24	110 - 120
100	62.2	91.2	12.2	12	12	100 - 110
90	55.9	82	10.9	10	54	90 - 100
80	49.7	72.9	9.7	9	42	80 - 90
70	43.5	63.8	8.5	8	30	70 - 80
60	37.3	54.7	7.3	7	18	60 - 70
50	31.1	45.6	6.1	6	6	50 - 60
40	24.9	36.5	4.9	4	54	40 - 50
30	18.6	27.3	3.6	3	36	30 - 40
<30	<18.6	<27.3	<3.6	<3	36	<30

Figure 2.8/ Speed Conversion Table

2.4/ Lane Proportioning Constants

In order to provide the average lane information as in the 'PCS5' series of routines, a known contribution, from each lane, based on the number of lanes, must be made. These contributions are derived from the Lane Proportion Constants. Their derivation can be seen below:

PCLN1	0	For 1 Lane
PCLN2	128	2 Lanes = (256-128)/256 = 128/256 = 0.50
PCLN3	171	3 Lanes = (256-171)/256 = 85/256 = 0.33
PCLN4	192	4 Lanes = (256-192)/256 = 64/256 = 0.25
PCLN5	205	5 Lanes = (256-205)/256 = 51/256 = 0.20
PCLN6	213	6 Lanes = (256-213)/256 = 43/256 = 0.17
PCLN7	219	7 Lanes = (256-219)/256 = 37/256 = 0.14
PCLN8	224	8 Lanes = (256-224)/256 = 32/256 = 0.13

3/ RAM PAGES

In the following section, a detailed illustration of the first five RAM pages will be given. It will deal with RAM in greater detail than Section 2.1 and include parameter manipulation, where applicable, as well as the physical location.

3.1/ Page ØØxy

As can be seen in Figure 3.1, this page is the storage area for general PCS parameters. It also has the communications area, lane configuration and timing data. The communication software is a separate entity, being independent of PCS operation. The timing information is controlled by an independent routine, while lane configuration is manipulated by the user. Figure 3.2 details the parameter location within the page.

3.2/ Page Ø1xy

As can be seen in Figure 3.3, Page Ø1xy is the main storage area for lane data. In Figure 3.4, the actual parameters stored are shown to comprise volume, occupancy rates, length and occupancy counts. The format for the RAM page is two direction data, with 16 lanes of 16 bytes of information per lane. Therefore, the rows represent a particular parameter for the lanes, while the column identifies the particular lane.

In the software, the base variable name is given as the address. All the base variables have been defined in the 'Equate' file (Appendix A). Therefore, to step through the lanes, all that is needed is to address the lane variable in an indexed mode. This is exactly what happens. With the use of two routines, 'SETUP' and 'NEXTLN,' the index value is set to access the highest lane. Then the index is decremented by $1Ø each time, in order to step through the lanes in ascending order. The process is the same for both directions, which are treated separately, except for the Direction 2 variables which have a $8Ø offset to locate them properly in memory.

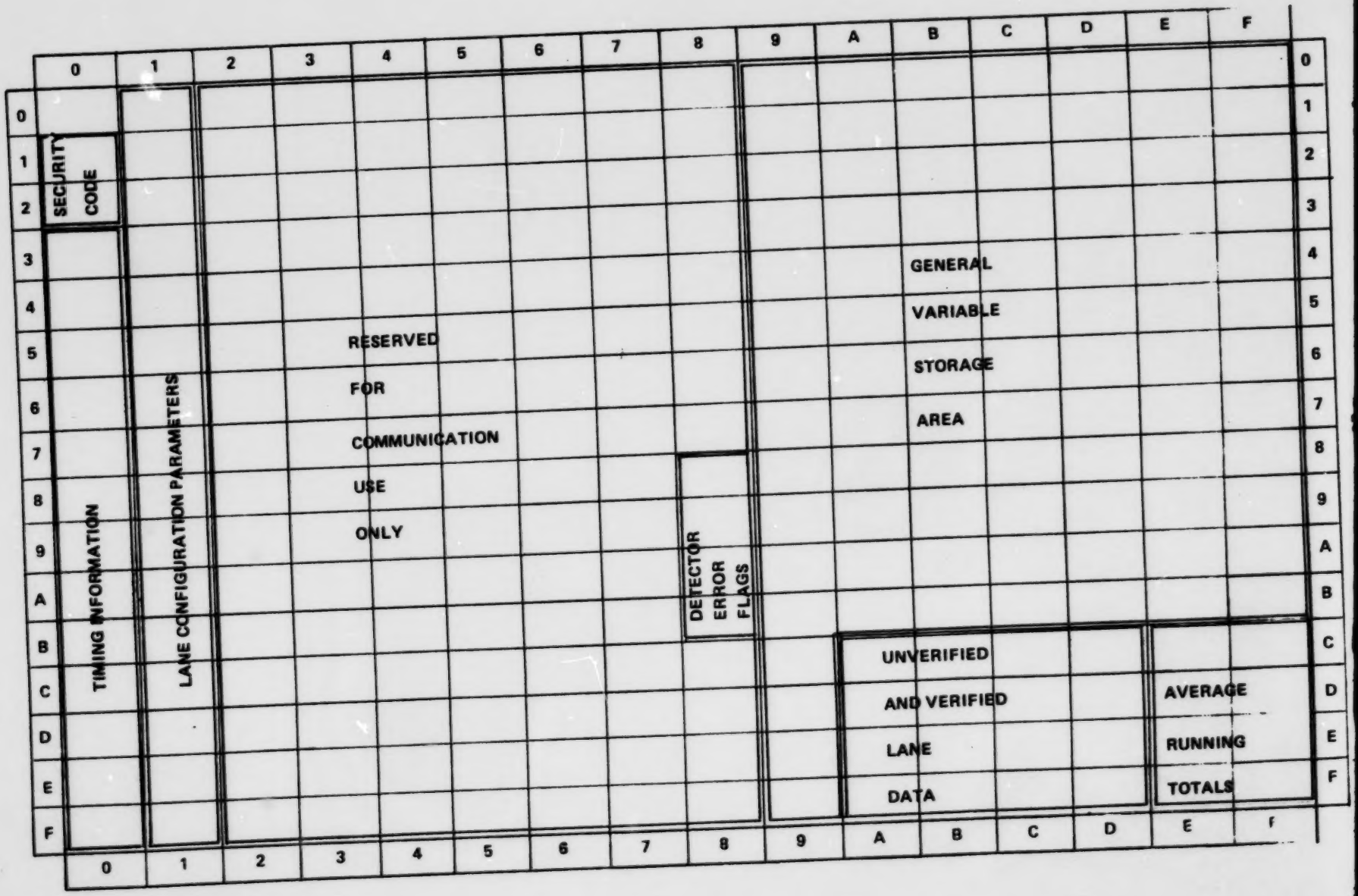


Figure 3.1/ General RAM Page $\phi\phi xy$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0		NLANE1						VOLDAY			SPC1	GO	DTCTRM	DTCTRL			0
1		NLANE2						WW		VODAY1	SPC2	GOT					1
2		LDESW1								VODAY1+1	KAY2	GOA				XLANE	2
3	YEAR	LDESW2						VOBUF1			TRUCK	GOTA			COUNT1	PCW1	3
4	MONTH	NVOL1						VOBUF1+1			LENGTH				COUNT2	KAY1	4
5	DAOFMO	NVOL2						SPDDAY		SIZE	LENGTH+1				COUNT3	TEMPϕ1	5
6		VOLDW1						SPDAY1		PTRL	SPEED				COUNT4	TEMPϕ2	6
7		VOLDW1+1						SPDAY1+1		PTRL+1	SPEED+1				COUNT5	TEMPϕ3	7
8		VOLDW1+2						SPBUF	DETER1	PTRS	VODAYL				COUNT6	TEMPϕ4	8
9	DAOFWK	VOLDW1+3							DETER2	PTRS+1	VODAYL+1					TEMPϕ5	9
A	HOUR	VOLDW1+4						LENDAY	DETER3		VODAYS				INDEX	TEMPϕ6	A
B	MINUTE	VOLDW1+5						LNDAY1	DETER4	OC1OCM	VODAYS+1				INDEX+1	TEMPϕ7	B
C	SECOND	VOLDW1+6						LNDAY1+1		OC1OCL	DATAϕM	DATAϕL			AR1MOM	→	C
D		VOLDW1+7						LNBUF1		PTR	DATA1M	DATA1L			AR1MOL	→	D
E		LDNW1					MRAM	LNBUF1+1		PTR+1	DETINM	DETINL			AR1MVM	→	E
F		LDNW2						DAYMAX		CYCLE	STCHGM	STCHGL			AR1MVL	→	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Figure 3.2/ Detailed RAM: ϕϕxy Organization

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0																	0
1																	1
2																	2
3																	3
4																	4
5		DIRECTION ONE									DIRECTION TWO						5
6		LANE DATA									LANE DATA						6
7																	7
8																	8
9																	9
A																	A
B																	B
C																	C
D		DIRECTION ONE									DIRECTION TWO						D
E		LONG VEHICLE COUNT PER LANE									LONG VEHICLE COUNT PER LANE						E
F	DIRECTION ONE OCCUPANCY PER LANE								DIRECTION TWO OCCUPANCY PER LANE								F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Figure 3.3/ General RAM Page ϕ1xy

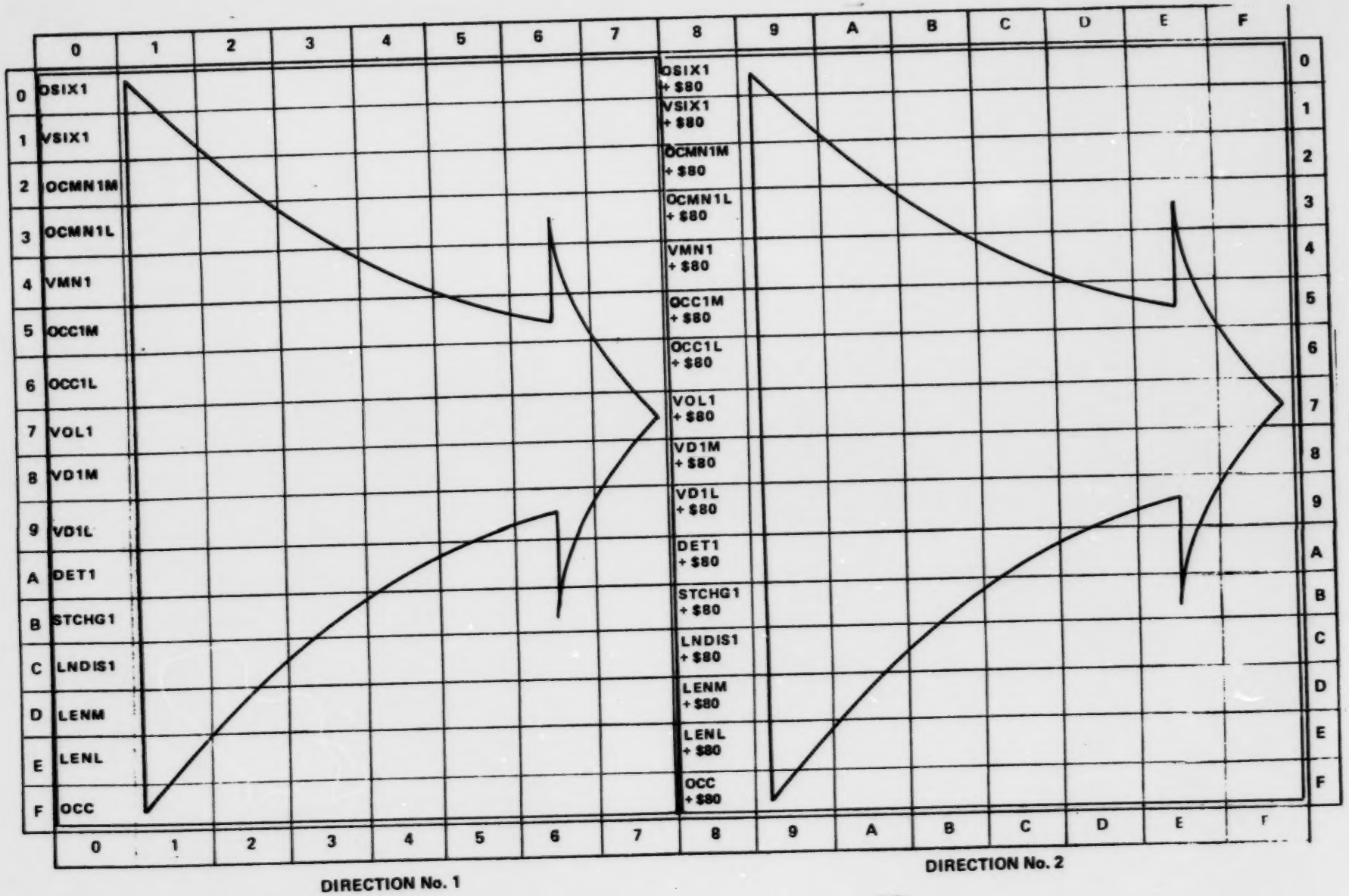


Figure 3.4/ Detailed RAM: ϕ1xy Organization

3.3/ Page Ø2xy

Figure 3.5, the general organization of the RAM, concerns itself with the interests of speed, calculation and classification. Figure 3.6 associates the parameters with the quantities of the page. The 'E' and 'S' suffixes of the transition timers and roll over flags represent a vehicle's end or start. The information contained in these and the loop characteristic storage area is accessed with index addressing. Here, as on Page 1, 'SETUP' and 'NEXTLN' index the base address to select the proper lane column.

The speed concerns on the page, 'SPD', 'REM' and 'SPED,' use the same type of addressing mode, except it has no association with the lanes. In this case, the base address is simply incremented by 1 in order to step the pointers down through the column of information.

3.4/ Page Ø3xy

This page of RAM is dedicated to the daily storage of classified speed. In the general RAM illustration (Figure 3.7), the layout can be seen for a five-day cycle. Figure 3.8 illustrates what would be found at the end of the day of initialization. It shows the parameters which make up the header, and it illustrates the order in which the classified values are stored in the transmit buffer.

As the days progress, this identical structure would appear for every completed day, rolling over onto Day 1 for Day 6. The layout of the speed classification in this buffer is more readily understood when considered in conjunction with Section 3.3.

3.5/ Page Ø4xy

From Figures 3.9 and 3.10, little insight is gained with respect to the structuring of the page. The reason for this is, as previously mentioned, the structure depends on the size and configuration of the station. But there are some general facts about the first page of the volume transmit buffer.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0				DIRECTION ONE							MSB	ISB	LSB	> 130			0
1				LANE TRANSITION TIMING BLOCK													1
2				DIRECTION TWO													2
3				LANE TRANSITION TIMING BLOCK													3
4				DIRECTION ONE LOOP SPACINGS													4
5				DIRECTION TWO LOOP SPACINGS					SPEED GROUP THRESHOLD	VALUES		SPEED					5
6				DIRECTION ONE LOOP DIMENSIONS								GROUP					6
7				DIRECTION TWO LOOP DIMENSIONS								DATA					7
8				DIRECTION ONE													8
9				COUNTER ROLL OVER FLAG													9
A				DIRECTION TWO													A
B				COUNTER ROLL OVER FLAG										< 30			B
C																	C
D																	D
E																	E
F																	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Figure 3.5/ General RAM Page ϕ2xy

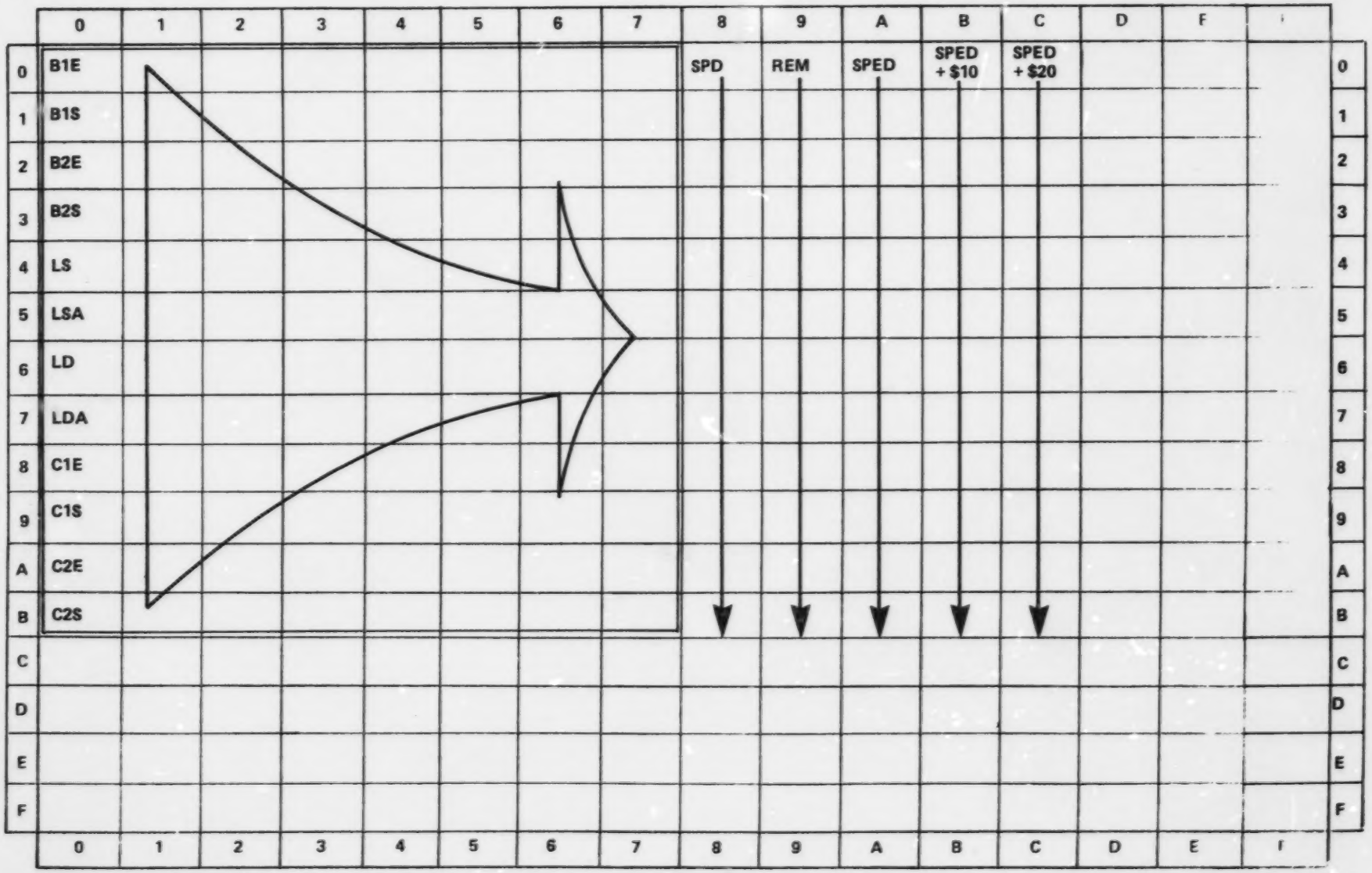


Figure 3.6/ Detailed RAM ϕ2xy Organization

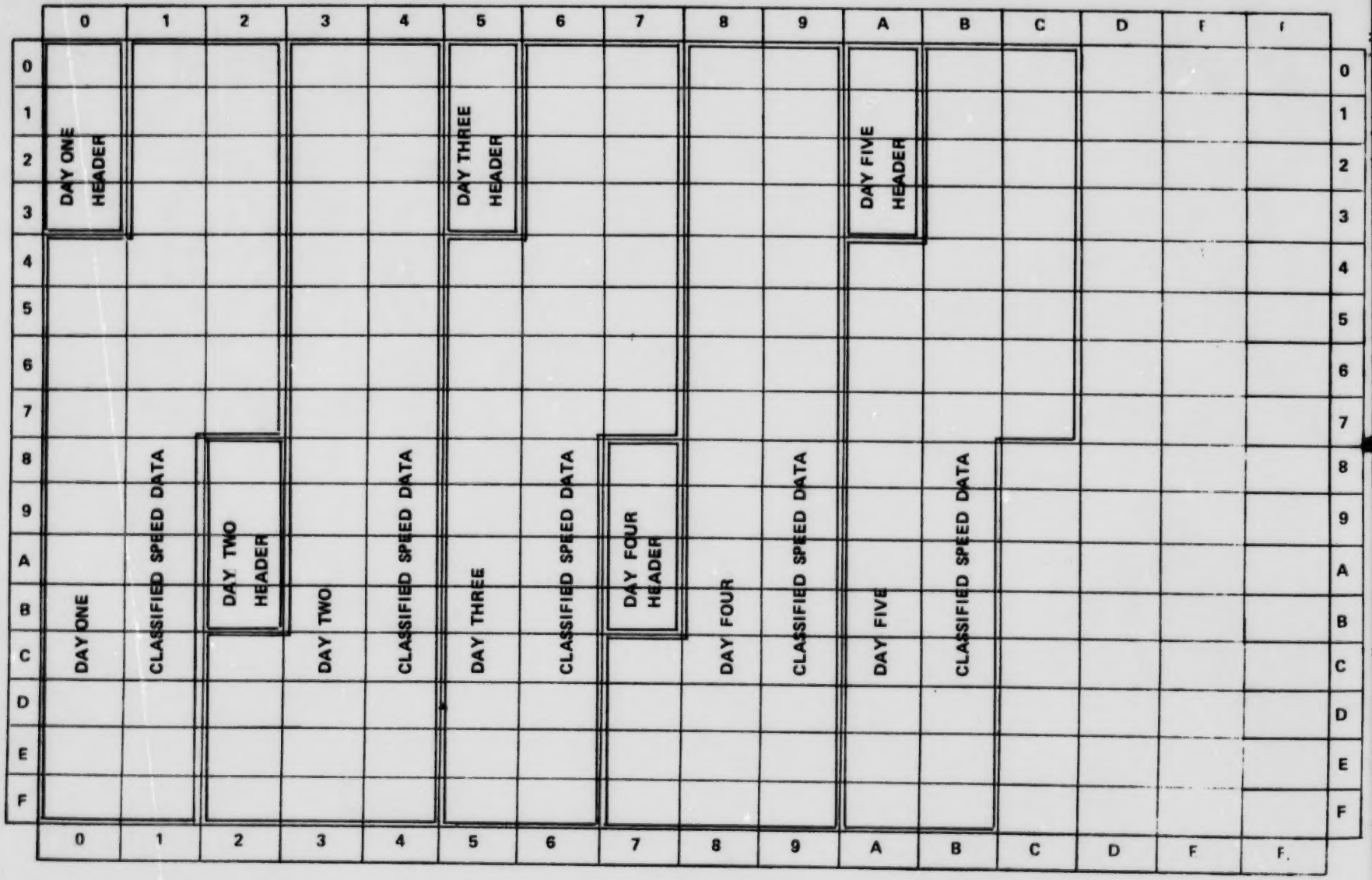


Figure 3.7/ General RAM Page ϕ3xy

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	YEAR	SPED,4	SPED + $10,9														0
1	MONTH	SPED + $10,4	SPED + $20,9														1
2	DAOFMO	SPED + $20,4	SPED,A														2
3	DAOFWK	SPED,5	SPED + $10,A														3
4	SPED	SPED + $10,5	SPED + $20,A														4
5	SPED + $10	SPED + $20,5	SPED,B														5
6	SPED + $20	SPED,6	SPED + $10,B														6
7	SPED,1	SPED +$10,6	SPED + $20,B														7
8	SPED + $10,1	SPED + $20,6															8
9	SPED + $20	SPED,7															9
A	SPED,2	SPED + $10,7															A
B	SPED + $10,2	SPED + $20,7															B
C	SPED + $20,2	SPED,8															C
D	SPED,3	SPED + $10,8															D
E	SPED + $10,3	SPED + $20,8															E
F	SPED + $20,3	SPED,9															F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Figure 3.8/ Detailed RAM ϕ3xy-Organization

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	DAY ONE VOLUME HEADER																0
1																	1
2																	2
3			DAY ONE														3
4			VOLUME														4
5			TRANSMIT														5
6			BUFFER														6
7																	7
8																	8
9																	9
A																	A
B																	B
C																	C
D																	D
E																	E
F																	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Figure 3.9/ General RAM Page ϕ4xy

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	YEAR																0
1	MONTH																1
2	DAOFMO																2
3	DAOFWK																3
4																	4
5																	5
6																	6
7																	7
8																	8
9																	9
A																	A
B																	B
C																	C
D																	D
E																	E
F																	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Figure 3.10/ Detailed RAM ϕ4xy Organization

The buffer always starts with its initial header at Ø4ØØ. It takes on the same type of format as seen in Ø3xy, with respect to the five-day cycle. Internally, though, the difference is dramatic. Two bytes are allocated for each value count sum, in each direction, which is determined by the user's configuration. But independent of the size, the Direction 1 sums appear first, followed by Direction 2.

4/ OPERATING PROCEDURES

Under normal operations, a user must define the physical layout of the loop detectors and impose the data handling format he/she requires. In starting up a new system, or when changes have been made to the system, a system initialization is also required. But, if the controller has just been powered down and back up again, system initialization is not required, since the battery backup maintains the value of RAM.

For example, suppose it was necessary to define a configuration of a situation as illustrated in Figure 4.1. The first step is to enter the Security Code Ø7,Ø5 into Addresses Ø1,Ø2. Assuming the time is correct, the next step is the definition of the configuration as in Figure 4.1.

Parameter	Address	Value
NLANE1	10	3
NLANE2	11	3
LDESW1	12	7
LDESW2	13	7
NVOL1	14	2
NVOL2	15	2
VOLDW1	16	3
VOLDW2	17	4
VOLDW3	18	3
VOLDW4	19	4
VOLDW5	1A	0
VOLDW6	1B	0
VOLDW7	1C	0
VOLDW8	1D	0
LDNW1	1E	7
LDNW2	1F	7

A detailed description of the parameter above can be found in Section 6.7.3. Basically, 'NLANE1/2' means that there are three lanes in each direction. The 'LDESW1/2' indicates which lanes are equipped with loop detectors. In the three lanes, there are two summed counts,

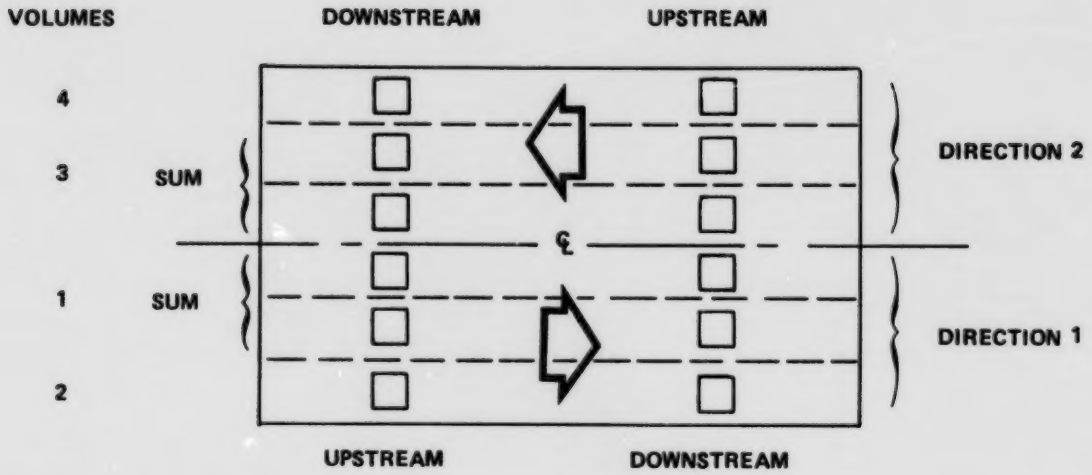


Figure 4.1/ Typical Lane Configuration

indicated by 'NVOL1/2,' and which lanes are associated for these counts is found in the 'VOLDW' series. It is very important that the unused portions of the 'VOLDW' series parameters, in Cases 5 through 8, be defined as zero. And last, there are the downstream description bytes found in 'LDNW1/2.' These entries followed by an initialization (any non-zero value in Ø6D) would start the system functioning as desired.

The only other variables which a user may like to change are the loop spacings 'LS' and loop dimensions 'LD.' Once the desired values (Section 2.3) are calculated, they may be entered in the appropriate memory locations (Section 3.3).

5/ ROUTINE SUMMARY

5.1/ Lane Data Pointer Initialization

Routine Request: 'SETUP'

Options: None

Function

Prior to accessing the lane and/or lane data area, the lane pointer must be reset to a datum. This is done by using this routine, which assigns the pointer a value of Ø17Ø. All lane information resides on Page Ø1, and Column 7Ø happens to be Lane 8 in Direction 1. Therefore, the various functions associated with a lane can be accessed in successive addresses of the appropriate column. Areas for Direction 2 are treated in a similar manner, with the addition of an $80 offset.

Comments

This is a very simple routine that serves a vital service. It should be kept in mind when examining the lane scanning provided by 'SETUP' and 'NEXTLN' that it is from the highest lane number in a particular direction to the lowest lane number.

5.2/ Lane Scan Control

Routine Request: 'NEXTLN'

Options: None

Function

If any portion of the software requires the lanes or lane data to be scanned, whether for input or data manipulation, this routine should be exercised. It will, when called, step through the lanes in descending order, highest to lowest, updating the address pointer INDEX, INDEX + 1.

As in the 'SETUP' description, lane data can be accessed from the base address, and the Direction 2 information requires the addition of an $80 offset.

Comments

Since the subroutine requires the use of Accumulator 'A', its contents, defined by the calling routine, are pushed into the stack at the outset of 'NEXTLN' and pulled before the subroutine relinquishes control. This ensures the error-free operation of the calling routine.

5.3/ 8-Bit Multiplication

Routine Request: 'MULT'

Options: None

Function

This routine multiplies the 8-bit unsigned number in Memory Location ØØF8 by the 8-bit unsigned number in Memory Location ØØF9, and then places the 8 most significant bits of the result in Memory Location ØØFA and the 8 least significant bits in Memory Location ØØFB.

Comments

To run this algorithm on a system with a 1 MHz clock takes between 190 and 230 μs.

5.4/ 8-Bit Division

Routine Request: 'DIV'

Options: None

Function

This routine divides the 8-bit unsigned number in Memory Location ØØF9 by the unsigned 8-bit number in Memory Location ØØF8. It stores the quotient in Memory Location ØØFA and the remainder in Memory Location ØØFB.

Comments

The algorithm takes between 170 and 210 μs to divide, with a 1 MHz clock.

Ensure that the divisor and dividend are correctly assigned, since it is the usual source of unexpected results.

5.5/ 16-Bit Number Comparison

Routine Request: 'CMP16'

Options: None

Function

This generalized comparison routine operates on 8 bits at a time. The most significant byte of the indexed number is compared to the number stored in Memory Location ØØF6, and unless the quantities are equal, the process can clear or set the carry bit to indicate the outcome. If they are equal, the least significant byte of the indexed number is compared to the value in Memory Location ØØF7. If the indexed value is greater than the memory location value, the carry bit is set; otherwise, it is cleared.

Comments

Remember that if the carry bit is clear, the values can be equal as well as the Memory Location ØØF6, ØØF7 being greater than the indexed value.

5.6/ Lane Proportioning Control

Routine Request: 'PRPRT'

Options: (a) 8-bit proportioning
(b) 16-bit proportioning

Function

The subroutine performs the operation N = N(1-PCW/256), where N represents the number to be proportioned, and PCW the proportion of 'N' to be subtracted. Upon completion, Accumulator 'A' holds the MSB proportion of 'N,' and Accumulator 'B' holds the LSB proportion.

Comments

The routine is capable of handling both 8- and 16-bit numbers determined by the index register.

5.7/ System Initialization

Routine Request: 'INIT'

Options: Lane configuration

Function

The purpose of this routine is three-fold. It is required to:

- clear memory,
- assign definitions and calculate variables, and
- output the first header.

It is desirable to clear memory in bulk since many of the parameters on Page Ø must be initialized to zero anyway. Also, it clears the transmit buffers to avoid getting erroneous data in the unused hours of the initial day.

Since the programs to follow require predetermined values, dealing with lane configuration and the like, the routine is the logical place for the calculation and the definition of parameters.

The third function, initial header output, naturally belongs here because it is not required in the cyclic nature of the normal operating routines.

Comments

To invoke the initialization routine, the Memory Address $Ø6D must be set to any non-zero value. When this is done and the 'INIT' routine completed, the value of $Ø6D is restored to zero.

It should be noted that the memory clearing portion has an arbitrary upper limit of $Ø7DF, which represents 2K bytes of memory minus the stack location. As a result, the routine will not clear transmit buffers which occupy addresses higher than this location, which is not critical, but undesirable.

5.8/ Speed Output and Header Control

Routine Request: 'HEADER'

Options: None

Function

For the purpose of identifying a particular day's data, it is preceded in memory by a header block. This consists of 4 bytes as is appropriate for the particular day's data:

1/ current calendar year,
2/ current calendar month,
3/ day of the month, and
4/ day of the week.

Along the same lines, a myriad of pointers must be updated, by the routine, to keep header data, days of data, cycle and position in the circular buffer valid.

'HEADER' is also called upon to output the speed data into its transmit buffer. Since speed is only provided once a day, and 'HEADER' is run once a day, their association is natural.

Comments

'HEADER' is called as a 24 h routine. Values in the header block are represented in BCD.

5.9/ Detector Input Scan Control

Routine Request: 'INPUT'

Options: None

Function

As its name implies, this routine is responsible for the acquisition of detector signals. It does so by providing the correspondence between hardware detector assignments, port and bit numbers, and internal software detector assignments.

Comments

The 'INPUT' routine is called in the 1/60 of a second time slot by 'PCS2.'

5.10/ Single Error Correction Routine

Routine Request: 'PCS2'

Options: None

Function

This process screens incoming raw data by checking for valid vehicle transition data sequences. By considering three sequential data sets, valid and spurious data can be separated and indicated through state change information. The result of the exercise is that verified data are ready for further processing.

Comments

The correction routine is called every 1/60 of a second.

5.11/ Upstream Detector Servicing Routine

Routine Request: 'PCS3'

Options: None

Function

At the point of initiating the 'PCS3' series of programs, valid data have been established and are now ready to be acted upon. This routine, when appropriate, adjusts volume counts, starts vehicle transition timing and secures occupancy for speed and length routines.

Comments

The routine is executed at a call rate of 1/60 of a second.

5.12/ Downstream Detector Servicing Routine

Routine Request: 'PCS3B'

Options: None

Function

This version of the 'PCS3' series of programs is responsible for indicating that timing data for a vehicle are ready for processing. This is done by manipulating the transition timing flags, which is the sole responsibility of the downstream detectors.

Comments

The routine is executed at a call rate of 1/60 of a second.

5.13/ Lane Data Handling Routine

Routine Request: 'PCS4'

Options: None

Function

The 6 s volume/occupancy counts are added to the 1 min and hourly counts by this routine. In the case of hourly volume counts, the routine derives absolute counts as well as running totals. Occupancy, on the other hand, is not compiled in hourly format.

Comments

For understanding of the running total data, the routine 'PRPRT/PRPRT8' should be digested, as 'PCS4' relies on it very heavily.

The execution rate of this data handling routine is 6 s.

5.14/ Average Running Total Routine

Routine Request: 'PCS5'

Options: Number of lanes for average

Function

This very straightforward routine simply takes the running totals for all the eligible lanes and averages them over the number of lanes desired.

Comments

These values generated have no application in Phase II software as it stands. But in case PCS is required to perform some control features in the future, it was thought to be more appropriate to incorporate these values during its development rather than later.

This routine functions at a execution rate of 6 s.

5.15/ Detector Failure Check Routine

Routine Request: 'PCS6'

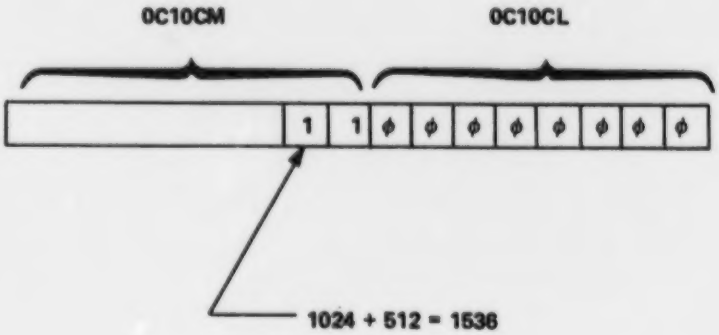
Options: Occupancy limit threshold

Function

This routine checks the lane's detector status. Only a simple test is conducted on the incoming signal from the loop detectors and their amplifiers. Although the routine has provisions to expand to many more failure checks, the only one present now is if the lane appears to be occupied greater than a pre-defined time. The threshold limit 'OC1OCM/OC1OCL' is located in $Ø9B/Ø9C and defined in 'INIT.'

Comments

The default value is set for approximately 43% of the 1 min occupancy (Figure 5.1).



SINCE: MAXIMUM OCCUPANCY FOR 1 MIN IS 60 X 60 = 3600

THEREFORE: 1536 ÷ 3600 x 100 = 42.6%

Figure 5.1/ Occupancy Limit Threshold

5.16/ Hourly Data Transfer Routine

Routine Request: 'PCS7'

Options: None

Function

The information concerning lane volumes is transferred to the transmit buffer with the use of this routine. It calculates the appropriate time slot and then inserts the volume data, summed where required, for both directions, into the appropriate day of the transmit buffer.

Comments

The long vehicle counts are treated identically to the volume counts described above and are handled by 'PCS7A.'

5.17/ Vehicle Speed and Length Classification

Routine Request: 'PCS8'

Options: None

Function

Data which have been received from the lanes are manipulated to calculate the vehicles' speed and to determine whether or not the vehicles exceed 6.5 m in length. In vehicle length, a vehicle either increments the count, or is excluded, whereas for speed, there are 12 speed ranges.

Comments

Long vehicle counts are loaded into the transmit buffer by 'PCS7A,' while speed is done on a daily basis by 'HEADER.'

6/ ROUTINE DESCRIPTION

6.1/ 'Setup'

6.1.1/ Program Listing Reference

The relevant program listing may be found in Appendix A, starting on Page 163, Line 26Ø, extending to Line 263.

6.1.2/ Program Explanation

Since the PCS system is capable of scanning up to 16 lanes and does so on a constant basis, a need was perceived for a routine to initialize the data pointer. The function is required in 15 different locations throughout the software, which reinforced the need for the subroutine 'SETUP.'

The routine starts by defining its origin to be at CØ6Ø. Then the index register is loaded immediately with the hex value of Ø17Ø. The 2-byte assignment defines the RAM Page Ø1 as well as the seventh column which is seen in the second byte 7Ø. As can be seen in Section 3.2, this sets the lane data pointer at base of highest lane. With this datum established, indexed addressing techniques are employed to move the pointer down a particular column. Different columns are chosen by exercising another subroutine, 'NEXTLN,' which will be dealt with subsequently.

This routine is terminated by the return from subroutine statement which, as its name implies, returns control to the routine which originally was called 'SETUP.'

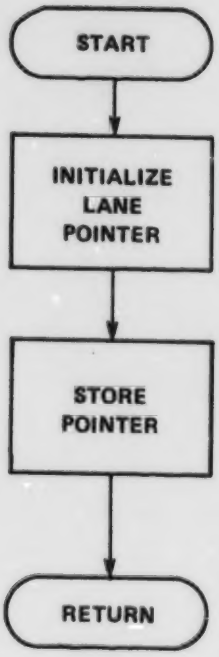
6.1.3/ Parameter Index

Routine: 'SETUP'

Description: Initializes lane data pointer.

Parameter	Description
INDEX	- this parameter along with INDEX+1 stores the 16-bit code used by the index register to allow accessing of lane data.

6.1.4/ Flow Chart



6.2/ 'NEXTLN'

6.2.1/ Program Listing Reference

The relevant program listing may be found in Appendix A, starting on Page 163, Line 240, extending to Line 252.

6.2.2/ Program Explanation

After calling the subroutine 'SETUP,' the lane pointer is defined to indicate the highest lane, which is Number 8, in Memory Location Ø17Ø. The subroutine 'NEXTLN' allows various routines to sequentially step down through the lanes and their data storage areas. This capability is required in 17 different locations throughout the software; therefore, 'NEXTLN' is justifiably an independent subroutine.

First, the origin of CØ3Ø is defined. Then the value of Accumulator A is saved. This precaution is taken because the routine requires the use of the accumulator and doesn't know its status with the main calling routine. Therefore, to avoid any possible errors, the subroutine saves the value in the stack and restores it to Accumulator A before returning from the subroutine. With the value stored, the routine sets about modifying the lane pointer.

First, the least significant byte of the current value is loaded into the accumulator. If the test to see if it is already the last lane fails, then a value of 10 hexidecimals is subtracted from it. Such a subtraction effectively shifts the pointer to the next lane data column in descending order. Having completed this, the new, least significant byte of the pointer is stored in INDEX+1, and then the complete new pointer is loaded into the index register.

Routines which use this subroutine test the carry bit to determine whether a complete set of eight lanes has been considered. So, if the series hasn't been completed, the last lane flag (carry bit) is cleared. If the last lane test described earlier is true, the flag is set by setting the carry bit. In either case, both alternatives lead to return from subroutine statements, which engages the calling routine.

In normal operation, the calling routine will arrange a loop including a JSR 'NEXTLN' statement. While a test of the clear carry bit is true, it will again call 'NEXTLN.' In this manner, all eight lanes in one direction are accessed in sequentially-descending order. When the carry bit is set by the previous call of 'NEXTLN' to indicate it has assigned the last lane address, the main routine will branch out of the loop and continue on with its other functions.

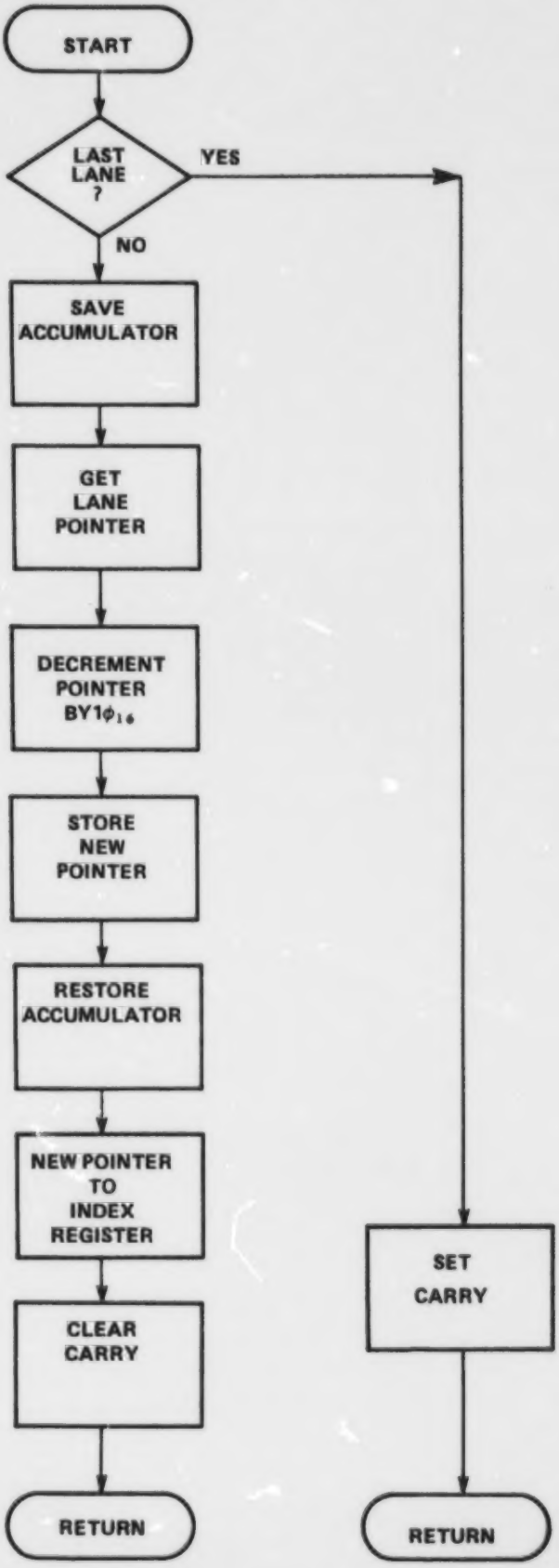
6.2.3/ Parameter Index

Routine: 'NEXTLN'

Description: Lane scan control.

Parameter	Description
INDEX+1	- the least significant byte of the index register storage area. Defines lane and data columns.
LSTLN	- a label used as the destination of a branch statement.
INDEX	- the most significant byte of the index register storage area. Defines the RAM page.

6.2.4/ Flow Chart



6.3/ 'MULT'

6.3.1/ Program Listing Reference

The relevant program listing may be found in Appendix A, starting on Page 165, Line 381, extending to Line 396.

6.3.2/ Program Explanation

The operation for performing multiplication here is the same as performing it by long hand. Since the numbers are binary, the only choice is whether to multiply by zero or 1; multiplying by zero obviously gives a zero result, while multiplying by 1 produces the same number that you started with in the multiplicand. Therefore, binary multiplication is reduced to the following.

- If the current bit in the multiplier is 1, add the multiplicand to the partial product.

- Ensure that numbers line up correctly by:
  - shifting the multiplier left 1 bit so that the bit to be examined is placed in the carry, and

  - shifting the product left 1 bit so that the next addition is lined up correctly.

The subroutine starts in the typical manner by defining the origin which, in this case, is C9BØ. Next, the accumulators are cleared, which effectively clears the product. The index register is loaded with the value eight, which dictates the number of iterations. Then the product and multiplier (TEMPØ4) are both shifted to the left. This procedure shifts the muliplier into the carry, and if the carry bit is 1, the product is added to the multiplicand (TEMPØ5) to form the new product. Once this is done, the counter, index register in this case, is decremented, and it branches back for another iteration until all eight have been completed. Upon completion, the product is stored in temporary locations TEMPØ6 and TEMPØ7 for use by the calling routine, after the execution of the return from subroutine statement.

The entire process takes between 190 and 230 μs, depending on the number of 1 bits in the multiplier.

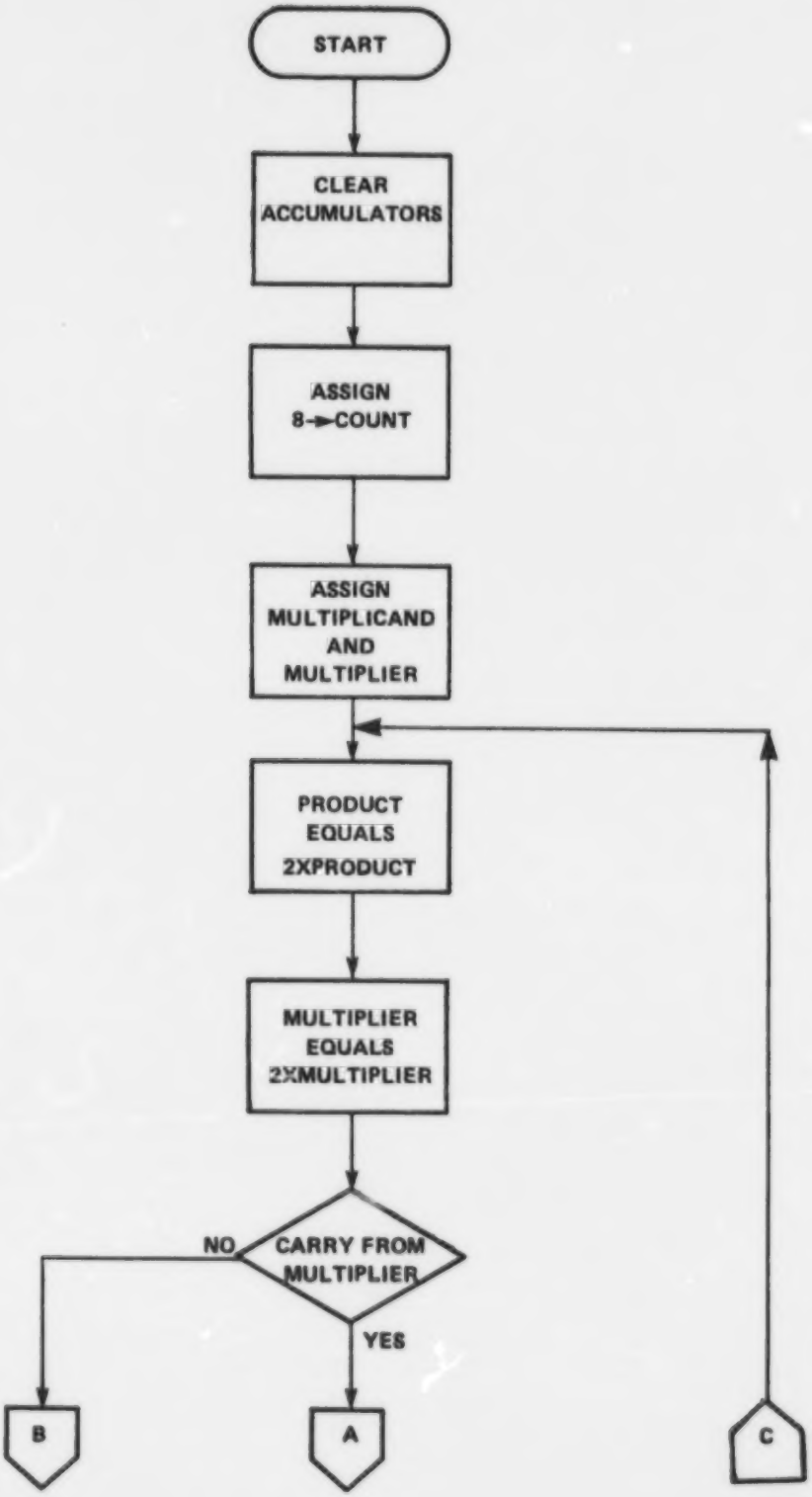
6.3.3/ Parameter Index

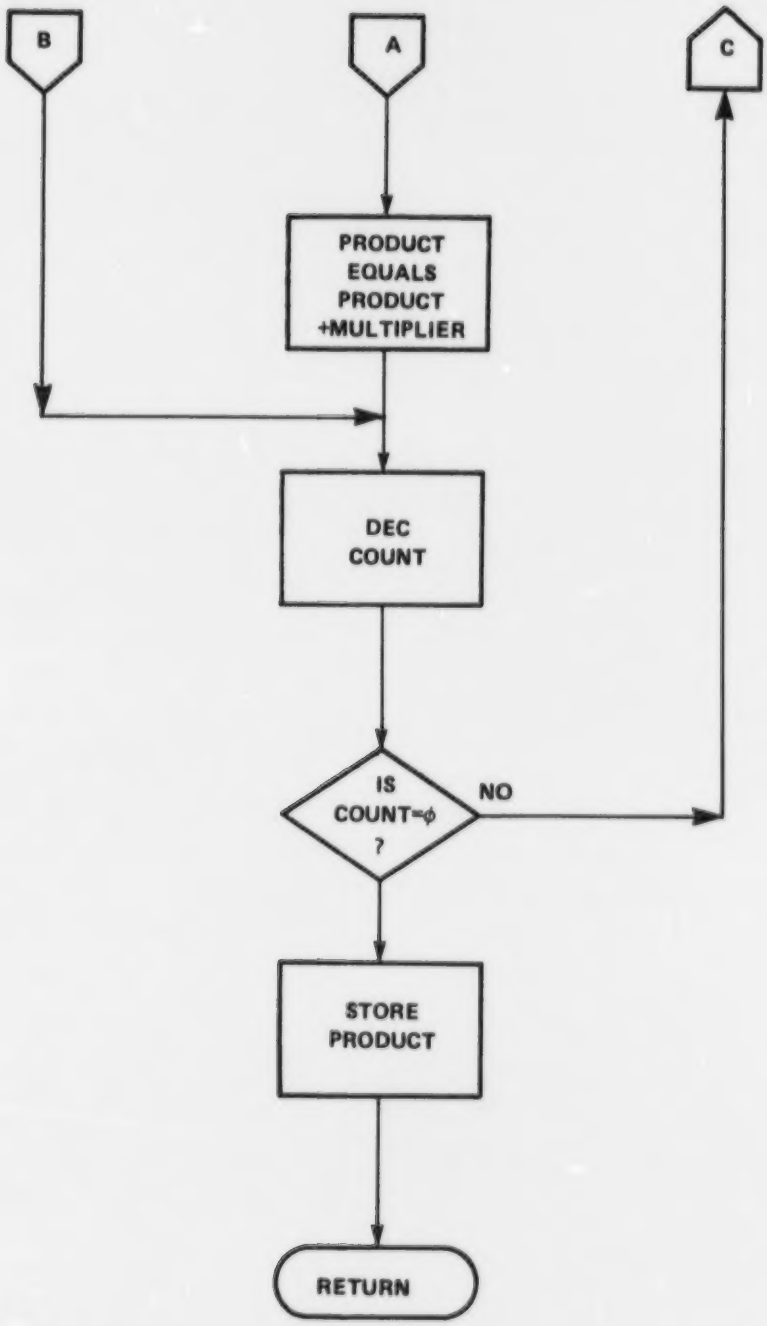
Routine: 'MULT'

Description: Provides for multiplication of two 8-bit numbers.

Parameter	Description
MM1	- a label for branch statement destination.
MM2	- a label for branch statement destination.
TEMPØ4	- a parameter used to store the multiplier.
TEMPØ5	- a parameter used to store the multiplicand.
TEMPØ6	- the most significant byte of the resulting product.
TEMPØ7	- the least significant byte of the resulting product.

6.3.4/ Flow Chart




B
A
C
PRODUCT EQUALS PRODUCT +MULTIPLIER
DEC COUNT
IS COUNT=ϕ ?
NO
STORE PRODUCT
RETURN

6.4/ 'DIV'

6.4.1/ Program Listing Reference

The relevant program listing may be found in Appendix A, starting on Page 165, Line 403, extending to Line 418.

6.4.2/ Program Explanation

The technique used for division in this routine is successive subtractions. Since the numbers are binary, the only question is whether the divisor can be subtracted from what is left of the dividend. If the divisor can be subtracted from the eight most significant bits of the dividend without a borrow, the corresponding bit in the quotient is 1; otherwise, it is zero.

After the familiar definition of origin, C9EØ in the case, the subroutine is initialized by setting the counter to eight and clearing the quotient. Next, the dividend (TEMPØ5) and quotient, temporarily in Accumulator 'A,' are shifted to line up properly. Then a trial subtraction is performed, and if there is no borrow, 1 is added to the quotient. After the counter is decremented, it is checked for zero, and depending on this outcome, the routine either branches for more iterations or stores the results. Upon completion, the quotient resides in TEMPØ6, and the remainder in TEMPØ7. In these locations, the calling routine can access the results when the subroutine's control is relinquished via a return from subroutine statement.

This subroutine takes between 170 and 210 μs to perform a division, depending upon the number of 1 bits in the quotient.

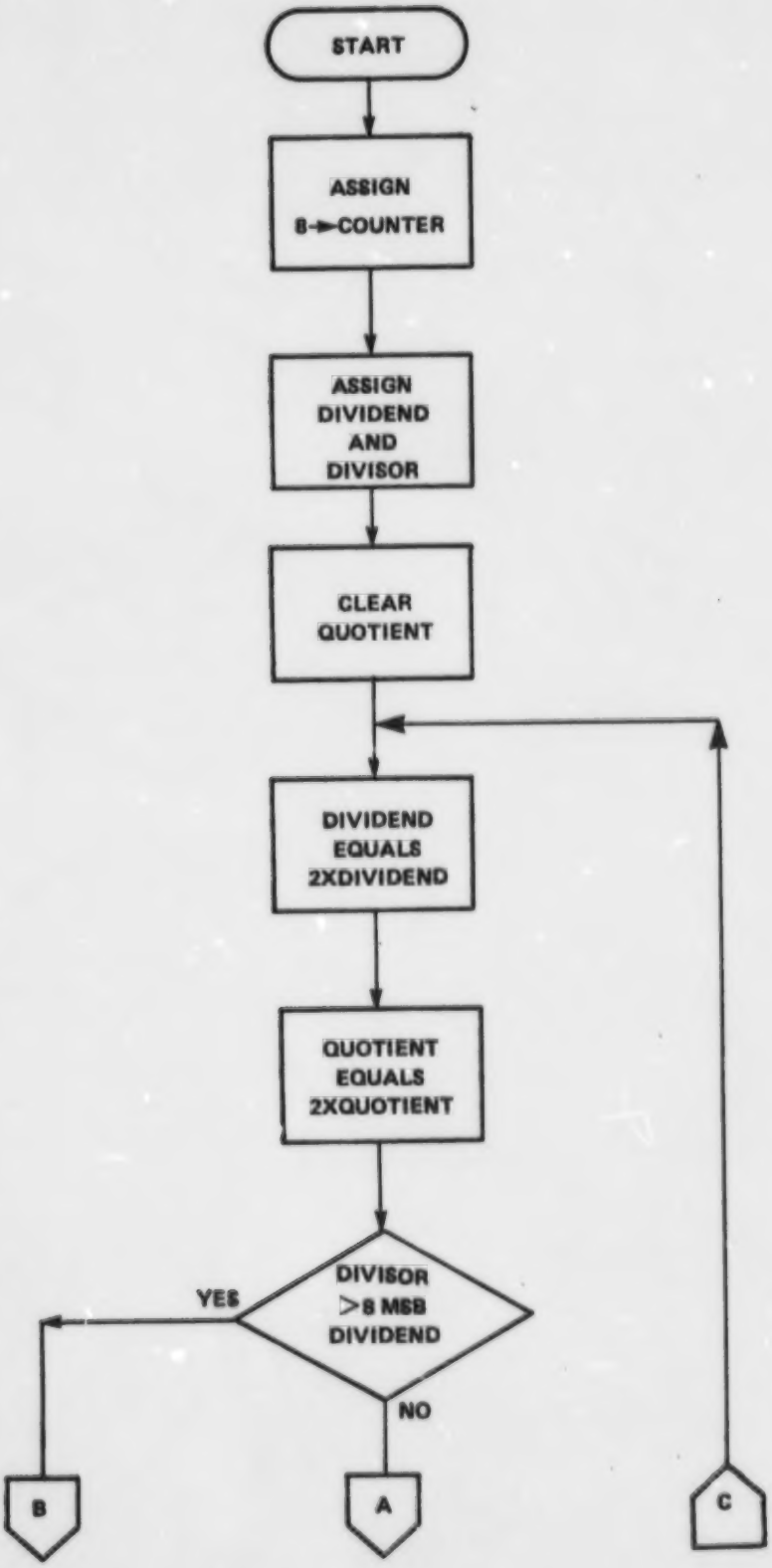
6.4.3/ Parameter Index

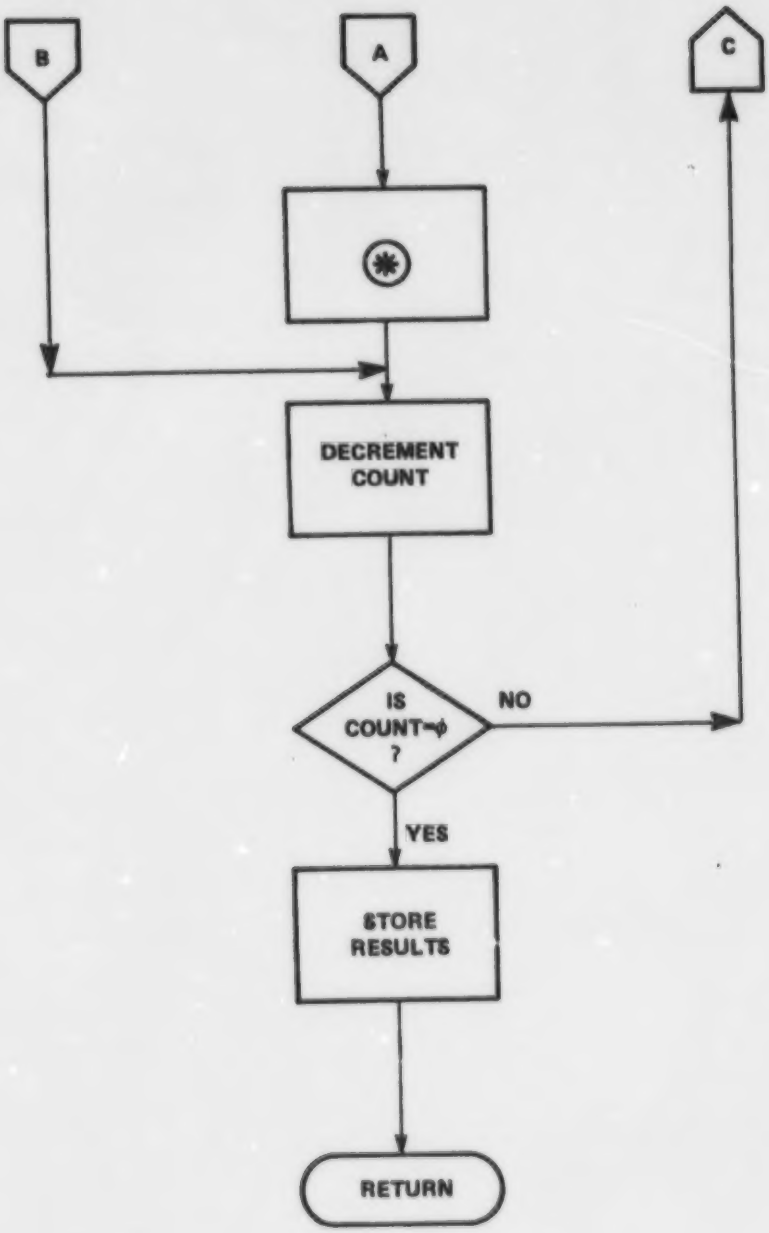
Routine: 'DIV'

Description: Provides for division of two 8-bit numbers.

Parameter	Description
MD1	- a label for branch statement destination.
MD2	- a label for branch statement destination.
TEMPØ4	- a variable for temporary assignment of the divisor.
TEMPØ5	- a variable for temporary assignment of the dividend.
TEMPØ6	- temporary storage area for the quotient.
TEMPØ7	- temporary storage area for the remainder.

6.4.4/ Flow Chart





(*) 8 MSBs OF DIVIDEND = 8 MSBs OF DIVIDEND - DIVISOR
QUOTIENT = QUOTIENT + 1

6.5/ 'CMP16'

6.5.1/ Program Listing Reference

The relevant program listing may be found in Appendix A, starting on Page 162, Line 219, extending to Line 234.

6.5.2/ Program Explanation

During the various computations involved in determining the PCS functions, it is necessary to compare 16-bit numbers. This subroutine is a generalized routine for doing such comparisons.

After the familiar introduction by defining the origin, CØØØ, the accumulator is stored to ensure error-free operation upon return to the calling routine. Then the MSB of one of the numbers is loaded into the accumulator using index mode. Next, it is compared to the MSB of the second number which has been previously asigned to TEMPØ2. If the indexed byte proves to be larger than TEMPØ2, then the carry bit is set; if less, the carry bit is cleared. On the other hand, if they prove to be equal, then the LSB is compared with similar results as above.

The results of the comparison are transferred to the calling routine via the carry bit. The carry bit will be set if the indexed 16 number is greater than that stored in TEMPØ2,Ø3. Otherwise, the carry bit will be cleared.

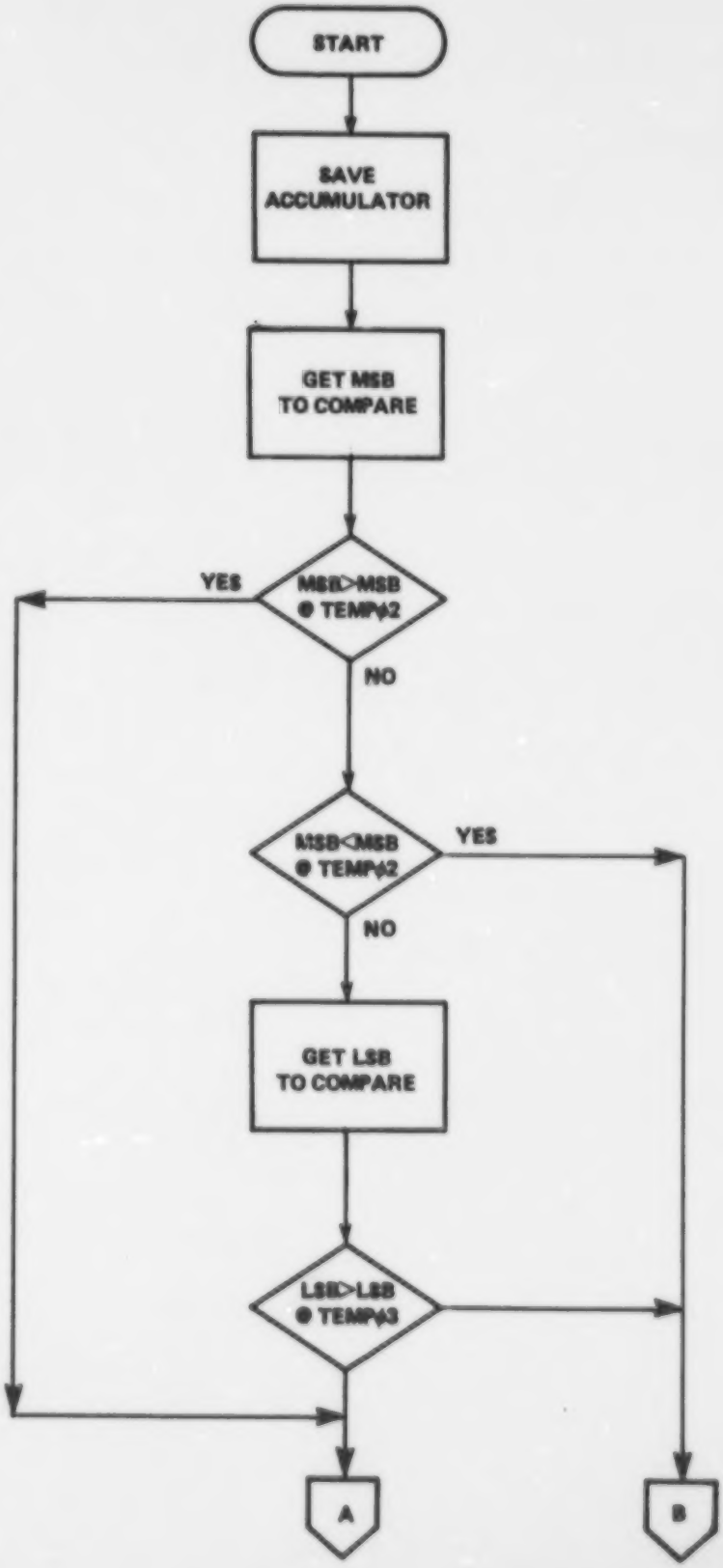
6.5.3/ Parameter Index

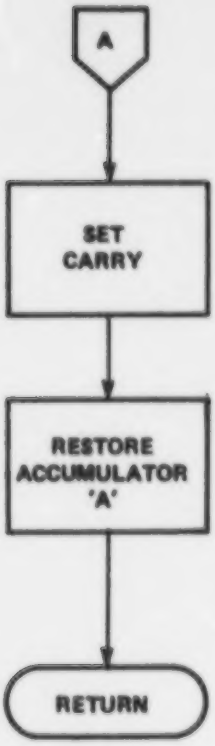
Routine: 'CMP16'

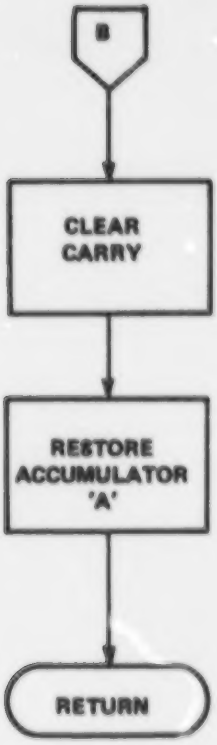
Description: A generalized subroutine to compare two 16-bit numbers.

Parameter	Description
TEMPØ2	- storage for the MSB of one of the numbers to be compared.
TEMPØ3	- storage for the LSB of one of the numbers to be compared.
ONEBIG	- the destination for a branch statement when the TEMPØ2,Ø3 variable is smaller.
ONESML	- the destination for a branch statement when the TEMPØ2,Ø3 variable is larger.

6.5.4/ Flow Chart




A
SET CARRY
RESTORE ACCUMULATOR 'A'
RETURN


B
CLEAR CARRY
RESTORE ACCUMULATOR 'A'
RETURN

6.6/ 'PRPRT/PRPRT8'

6.6.1/ Program Listing Reference

The relevant program listing may be found in Appendix A, starting on Page 164, Line 305, extending to Line 353.

6.6.2/ Program Explanation

1/ Theory[1]

For any number N:

$$(29/30)N = N - N(4/128) = N - 4(N/128)$$

and:

$$(9/10)N = N - N(13/128) = N - 13(N/128).$$

For a binary number, dividing by 128 corresponds to a shift of 7 bits toward the least significant digit. Then N(1 - 4/128) would be accomplished by shifting N right 7 bits and then subtracting the result from N four times.

Alternatively, the proportioning might be done as follows:

$$N - 4(N/128) = N - N/32$$

or simply a 5-bit shift towards the least significant digit. Using this method for proportioning 9/10 of a number:

$$N - 13N/128 = N - N(1/128 + 4/128 + 8/128)$$
$$= N - (N/128 + N/32 + N/16)$$

where dividing by 32 is a 5-bit shift and dividing by 16 is a 4-bit shift.

2/ Implementation

It will be noted that only one subtraction for each bit shift would be necessary for any proportion of 1/128. In the above example, a subtraction is made with the number shifted 4 bits, another shifted 5 bits and the third subtraction after shifting 7 bits. Since only one subtraction is made for each fractional part of the original number, a <u>binary proportion control word</u> can be set to vary the proportion of the original number the result is to be.

If each bit of the proportion control word represents an individual fractional part, any proportion can be represented in a binary coded, 7-bit word. Since 8 bits are available, fractional parts of 256 can be used.

---

[1]Developed by Caltrans District 11, 1979.

Proportion Control Word

Bit #	7	6	5	4	3	2	1	0
Fractional Part	1/2	1/4	1/8	1/16	1/32	1/64	1/128	1/256

An example of reproportioning a number to 9/10 of its original value follows. Since $9/10 = 1 - \frac{26}{256}$, the Bits Ø, 3 and 4 would be set in the proportion control word (PCW). The PCW can then be shifted left 1 bit at a time into carry. Whenever carry is set, the original number is shifted and subtracted from the original value. If the bit shifted to carry is not set, the original number is shifted only.

It should be noted that subtracting the largest fraction yields better resolution. For instance, suppose that the value to be reproportioned is 250, and the proportion wanted in the result is $9/10 = 1 - \frac{26}{256}$. The result should be about 224.61. Suppose 1/256 of the original value is subtracted 26 times. The fraction 250/256 is less than 1, so when a subtraction is made (250-Ø), the result remains 250. If, 2/256 of the original value is subtracted 13 times instead, the result would be 250 - 13, or 237. If the largest fractional parts are subtracted out in succession, i.e., 16/256, 8/256 and 2/256, the result is:

$$250 - 15 - 7 - 1 = 227$$

This is much closer; in fact it only varies 0.9% from the actual value.

Even better resolution can be obtained by 'rounding off' the successive subtractions. Suppose that when the number is shifted to obtain the fraction to be subtracted, the bit shifted to carry is set. This means that the portion shifted out of the word is at least 1/2 of the least significant bit. Then to round the subtraction off, the original word is decremented. In the above example:

$$250 - 15 - 1 - 7 - 1 - 1 = 225$$

Shifting the PCW left instead of right allows the value placed in it to be the number of 1/256 to be subtracted from the original value. For

example, if PCW - 10_{10} = 1010_2, the PCW will direct subtraction of 1/32(8/256) and 1/128(2/256) or 10/256 from the original value.

6.6.3/ Parameter Index

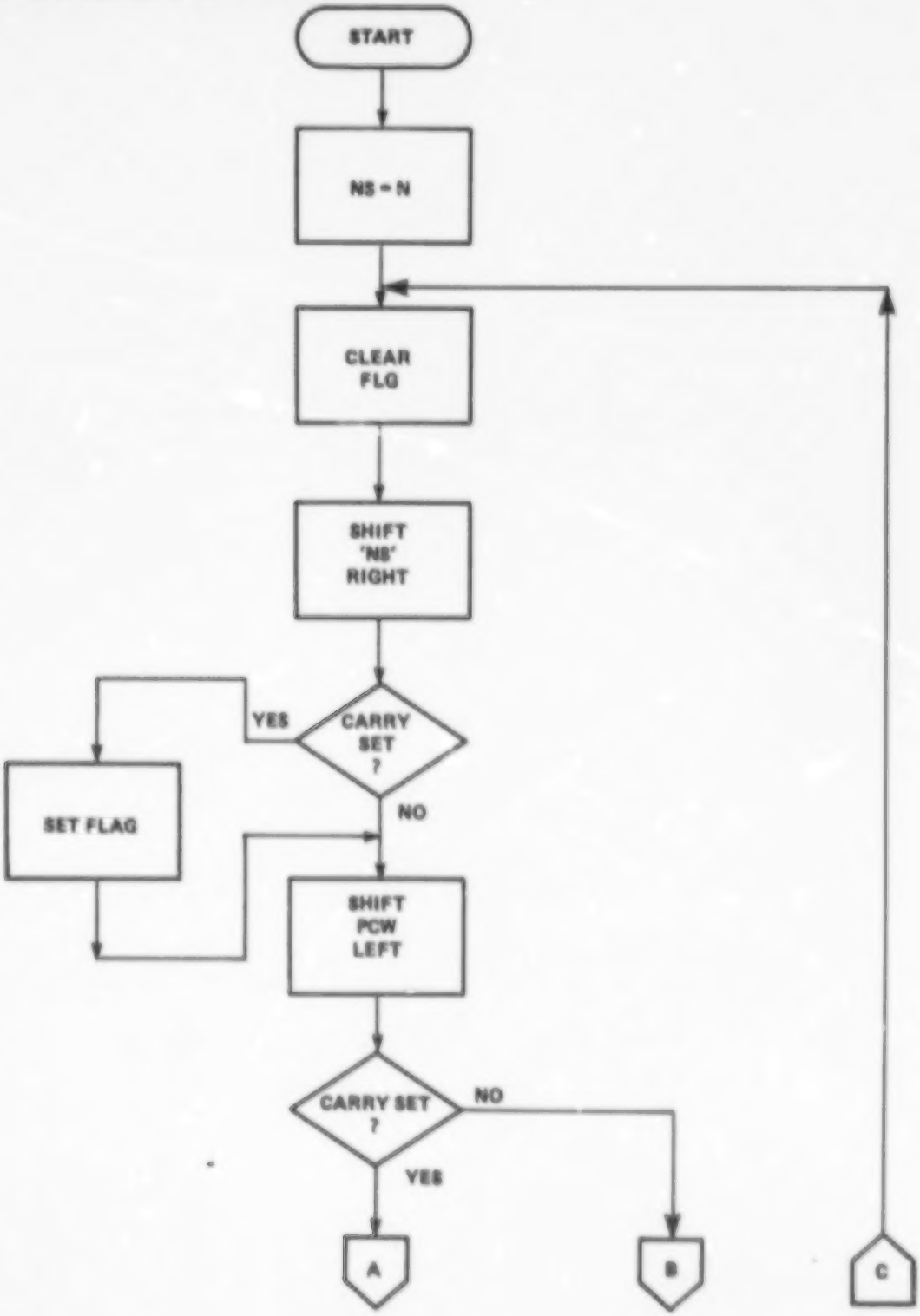
Routine: 'PRPRT/PRPRT8'

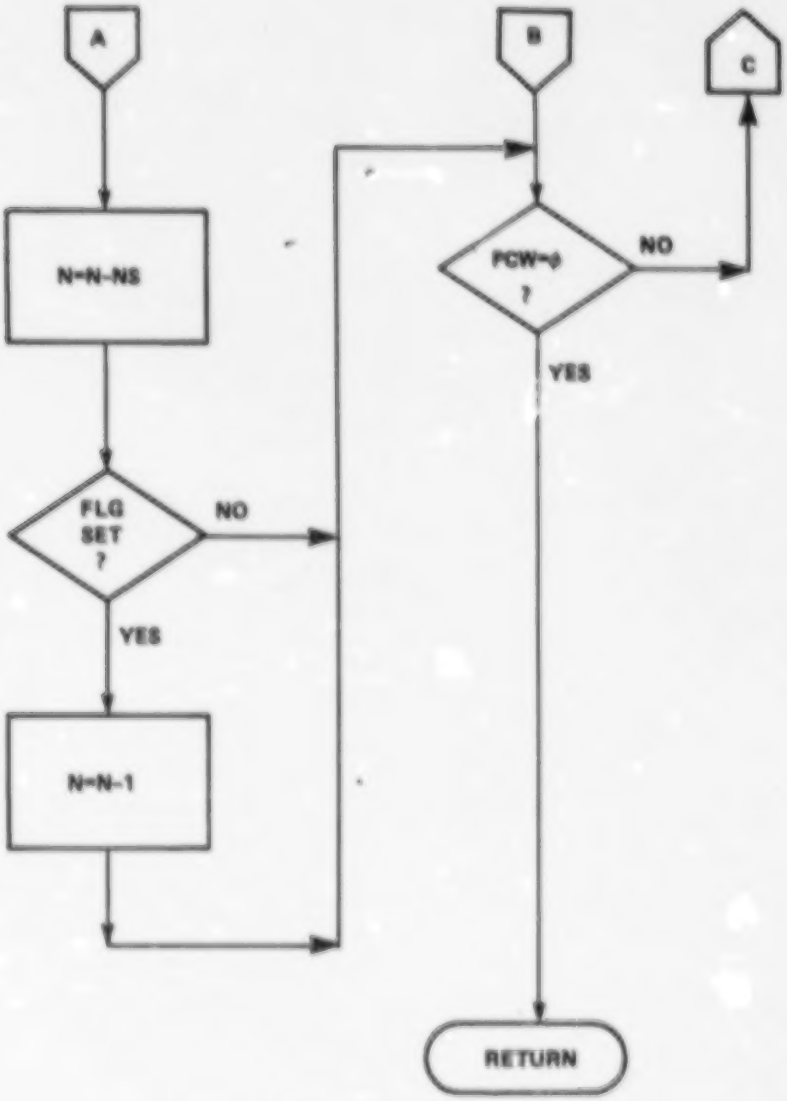
Description: Proportions 8 or 16 bits to provide pseudo running totals.

Parameters	Description
CHECK	- label for destination of branch statement.
AGAIN	- label for destination of branch statement.

6.6.4/ Flow Chart

N = ORIGINAL VALUE
NS = SHIFTED 'N'
FLG = ROUND OFF FLAG
PCW = PROPORTIONED CONTROL WORD




A
B
C
N=N-NS
FCW=Φ
?
NO
YES
FLG
SET
?
NO
YES
N=N-1
RETURN

6.7/ 'INIT'

6.7.1/ Program Listing Reference

The relevant program listing may be found in Appendix A, starting on Page 166, Line 436, extending to Line 725.

6.7.2/ Program Explanation

The initialization routine serves three functions:

1/ to zero memory,
2/ to define and calculate variables, and
3/ to output the first header.

Zeroing memory is rather a simple process. Basically it defines the boundaries in memory to be $88 and $7DF, and then steps through and clears addresses while incrementing the lower boundary condition. The lower boundary condition is set at $88 so as not to disturb the communications portion of Page Ø. The upper boundary is arbitrary.

Before getting into the section dealing with variable definitions, the lane disable variables are set. Scanning through 'LDNW1' downstream description byte for Direction 1 bit by bit, it is determined whether or not the lane is equipped with such a detector. If it is, in which case a '1' is found in that bit position, then the 'LNDIS1,' lane disable, function for that particular lane is cleared. Otherwise, the lane's binary bit position is entered as that of the lane disable parameter. This operation is repeated for Direction 2. When this is completed, it allows any of the lane and/or lane data scanning routines to determine whether any particular lane or data area is eligible for consideration.

The section for variable definition, which appears next, does as its name implies. Variables are transferred, cleared, initialized and generally conjured up. The function of any variable here may be found in the 'Parameter Index' (Section 6.7.3.). The latter two-thirds of the section deals with assigning the speed comparison and lane default values. Then the program gets into variable computation.

In order to store the various types of data, it must be determined how much space and, therefore, where the data pointers should be located. The process starts by counting the number of non-zero volume count descriptors and multiplying the sum by 2. This defines the space required for the storage of both value and long vehicle counts for 1 h, since each non-zero descriptor indicates a most and least significant

byte-sum to be stored. With the hourly requirements satisfied, the quantity is multiplied by 24 and stored as the length of the first day volume buffer (VOBUF1, VOBUF1+1). Since the length of buffer required per day is now known, a simple multiplication by 5 gives us the maximum storage requirements. This new value, offset by $0400, gives us the pointer to the start of the long vehicle count buffer. Next, in three distinctive groups, all the pointers are initialized, followed with output of the initial headers for volume, length and speed, before returning to the calling routine.

6.7.3/ Parameter Index

Routine: 'INIT'

Description: System initialization.

Parameter	Description
TEMPØ1	- temporary storage location used in varied applications.
TEMPØ2	- temporary storage location used in varied applications.
LDNW1	- downstream lane descriptor byte for Direction 1.
LDNW2	- downstream lane descriptor byte for Direction 2.
LNDIS1	- lane disable byte. A non-Ø value indicates the lane is unused.
DTCTRM	- MSB detector enable mask. (Bits match 'DETINM,' detectors; 1 allows controller to see detector.)
LDESW1	- upstream lane descriptor byte in Direction 1 (Bit Ø = 1 for Lane 1 detector, for example.)
LDNW2	- downstream lane descriptor byte in Direction 2.
DTCTRL	- LSB of detector enabling mask.

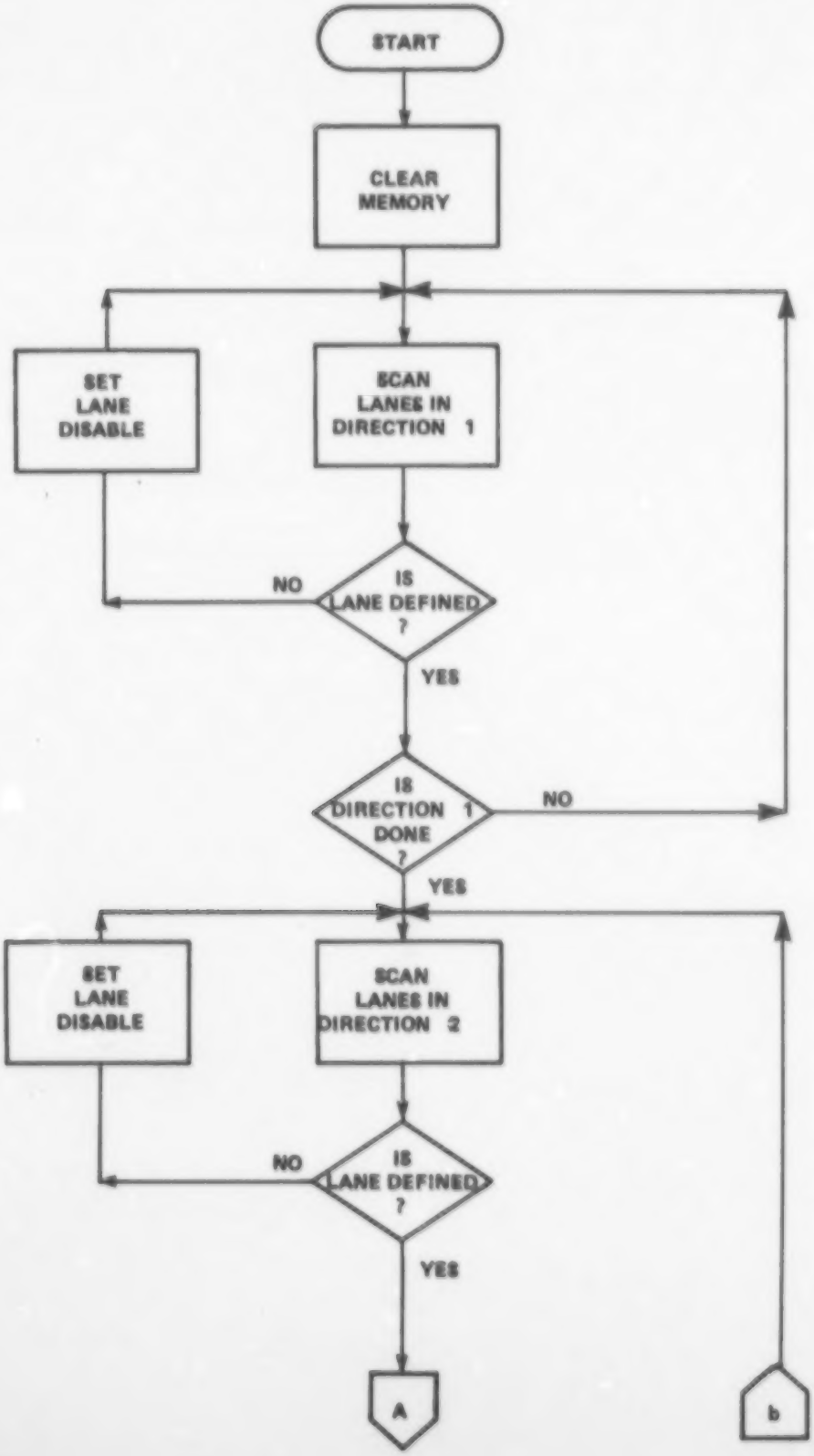
Parameter	Description
LDESW2	- upstream lane descriptor byte in Direction 2.
VOLDAY	- number of days volume data collected.
COUNT4	- a temporary storage location used in varied applications.
SPDDAY	- number of days speed data collected.
LENDAY	- number of days length data collected.
CYCLE	- a counter used to allocate address of day's pointer in circular transmit buffer.
KAY1	- flag to clear lane detector areas for Direction 1.
KAY2	- flag to clear lane detector areas for Direction 2.
PCW1	- proportion control word for 1 min data. (Set to '1A' [26/256] for 1 min).
VOBUF1	- length of an individual day's volume buffer.
SPDAY1	- pointer to the beginning of the speed buffer.

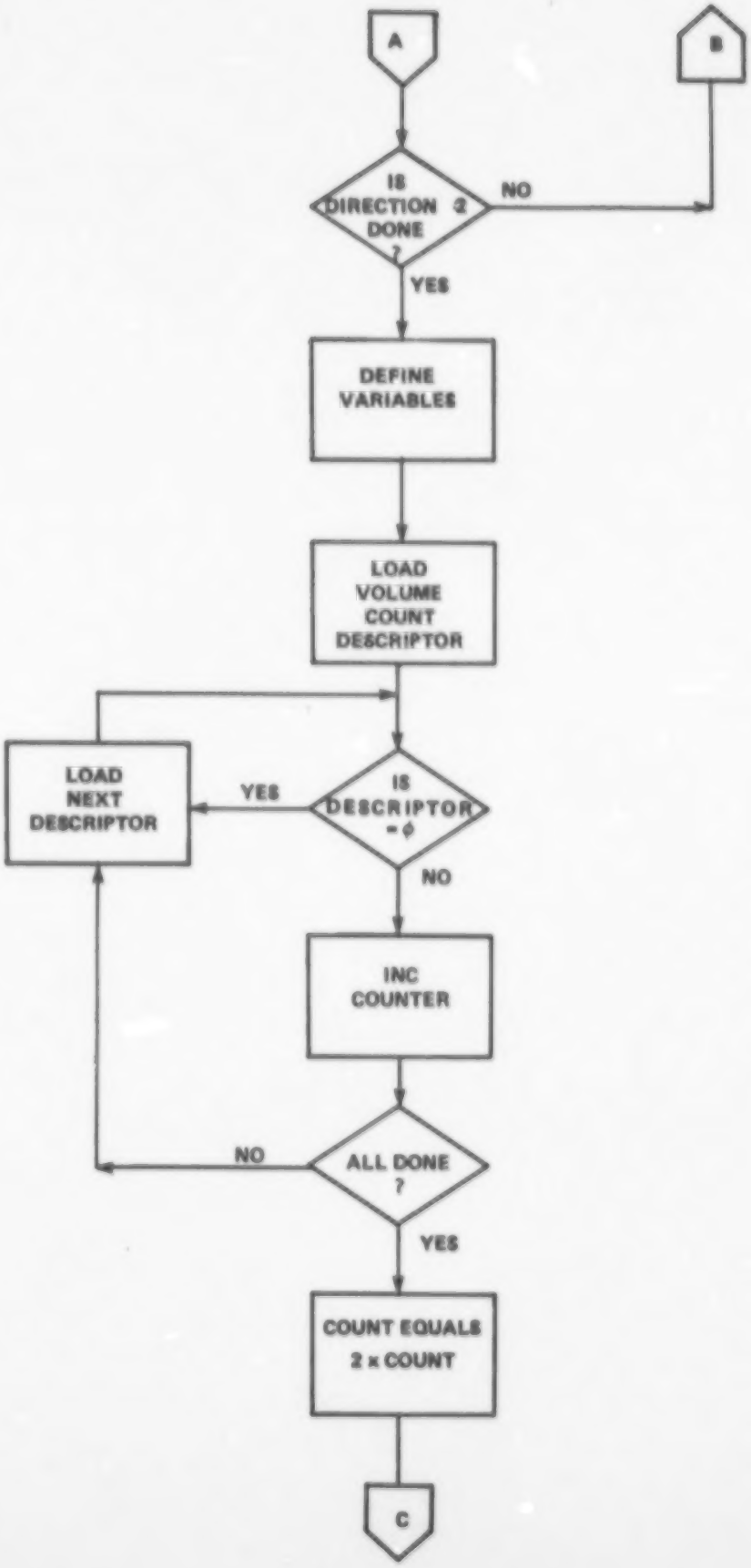
Parameter	Description
SPBUF	- length of an individual day's speed buffer.
OC1OCM	- error threshold value for 1 min detector occupancy.
SPEED	- initial speed buffer pointer at Ø3ØØ.
TRUCK	- threshold value for discriminating and classifying measured vehicle lengths.
DAYMAX	- the maximum number of days of data allowed within given memory size.
SPD	- base location for the MSB of converted speed classifications.
LS	- base location for individual lane's centre to centre loop spacing in Direction 1.
LSA	- base location for individual lane's centre to centre loop spacing in Direction 2.
LD	- base location for individual lane's loop dimension in Direction 1.
LDA	- base location for individual lane's loop dimension in Direction 2.
VOLDW1	- count descriptor byte for Volume 1.

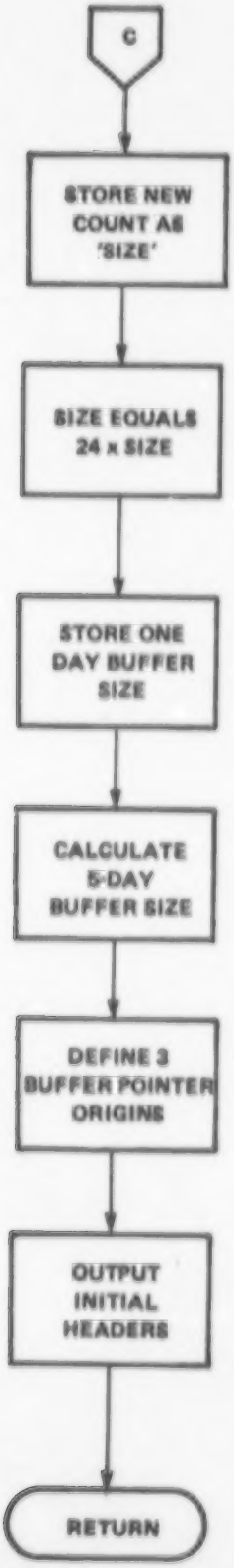
Parameter	Description
SIZE	- amount of memory space required to store all volume counts in both directions for 1 h.
MULT	- reference to the subroutine for multiplication.
TEMPØ3	- a temporary storage area used for varied applications.
TEMPØ4	- a temporary storage area used for varied applications.
TEMPØ5	- a temporary storage area used for varied applications.
TEMPØ6	- a temporary storage area used for varied applications.
TEMPØ7	- a temporary storage area used for varied applications.
LENGTH	- initial length buffer pointer.
LNBUF1	- length of an individual day's length buffer.
START	- initial volume buffer pointer at Ø4ØØ.
VODAY1	- pointer to beginning of the volume buffer.
WW	- same as 'VODAY1' but to be assigned a storage location in communications area.

Parameter	Description
PTR	- pointer to beginning of volume data.
VODAYL	- pointer to beginning of current day's length buffer.
LNDAY1	- pointer to beginning of length's transmit buffer.
PTRL	- pointer to beginning of length data.
VODAYS	- pointer to beginning of current day's speed buffer.
PTRS	- pointer to beginning of speed data.
YEAR	- current calendar year (BCD).
MONTH	- current calendar month (BCD).
DAOFMO	- current day of the month (BCD).
DAOFWK	- current day of the week (BCD).

6.7.4/ Flow Chart




A
B
IS DIRECTION 2 DONE ?
NO
YES
DEFINE VARIABLES
LOAD VOLUME COUNT DESCRIPTOR
LOAD NEXT DESCRIPTOR
YES
IS DESCRIPTOR = ϕ
NO
INC COUNTER
NO
ALL DONE ?
YES
COUNT EQUALS 2 x COUNT
C


C
STORE NEW COUNT AS 'SIZE'
SIZE EQUALS 24 x SIZE
STORE ONE DAY BUFFER SIZE
CALCULATE 5-DAY BUFFER SIZE
DEFINE 3 BUFFER POINTER ORIGINS
OUTPUT INITIAL HEADERS
RETURN

6.8/ 'HEADER'

6.8.1/ Program Listing Reference

The relevant program listing may be found in Appendix A, starting on Page 168, Line 746, extending to Line 919.

6.8.2/ Program Explanation

Prior to providing the headers on day changes, this routine looks after the transfer of speed information as well. It does so by indexed addressing of the 'SPED,' accumulated days speed information, and placing it in the speed transmit buffer. Since there are 12 classifications defined in the speed requirement, the process is repeated 12 times. It should be noted that 3 bytes are transferred for every speed group (format details in Section 3.3); hence, the 'SPED' address can be seen to be offset by two different hexidecimal values. Upon completion of the speed data transfer, the rest of the routine concerns itself with setting buffer pointers and providing headers.

The header information, and data for that matter, are arranged in circular buffers. This is accomplished by initializing the buffer pointer to a known location ('INIT' routine, Section 6.7.2) and then systematically incrementing the data and headers through memory until the maximum number of days to be stored is reached. Then the information overlays itself, generating the circular format. An illustration of the process may be seen in Figure 6.1. Recalculating the buffer pointers is straightforward, since the present position is always used and the buffer length required is known; therefore, the new pointers are the result of a simple summation. The variable 'CYCLE' keeps track of the number of days stored relative to the defined maximum, and when the limit has been reached, it directs the pointers to the beginning of the buffer for another cycle. This function should not be confused with 'VOLDAY' which keeps track of the number of days of data available at any given time. Under normal operation, 'VOLDAY' will increment daily up to the maximum number of days to be stored, retaining this value until the system is reset.

Once the buffer points have been established, it is a simple matter of reading the date time information and outputing these four functions in three different headers. Control is then granted to the calling routine with an RTS statement.

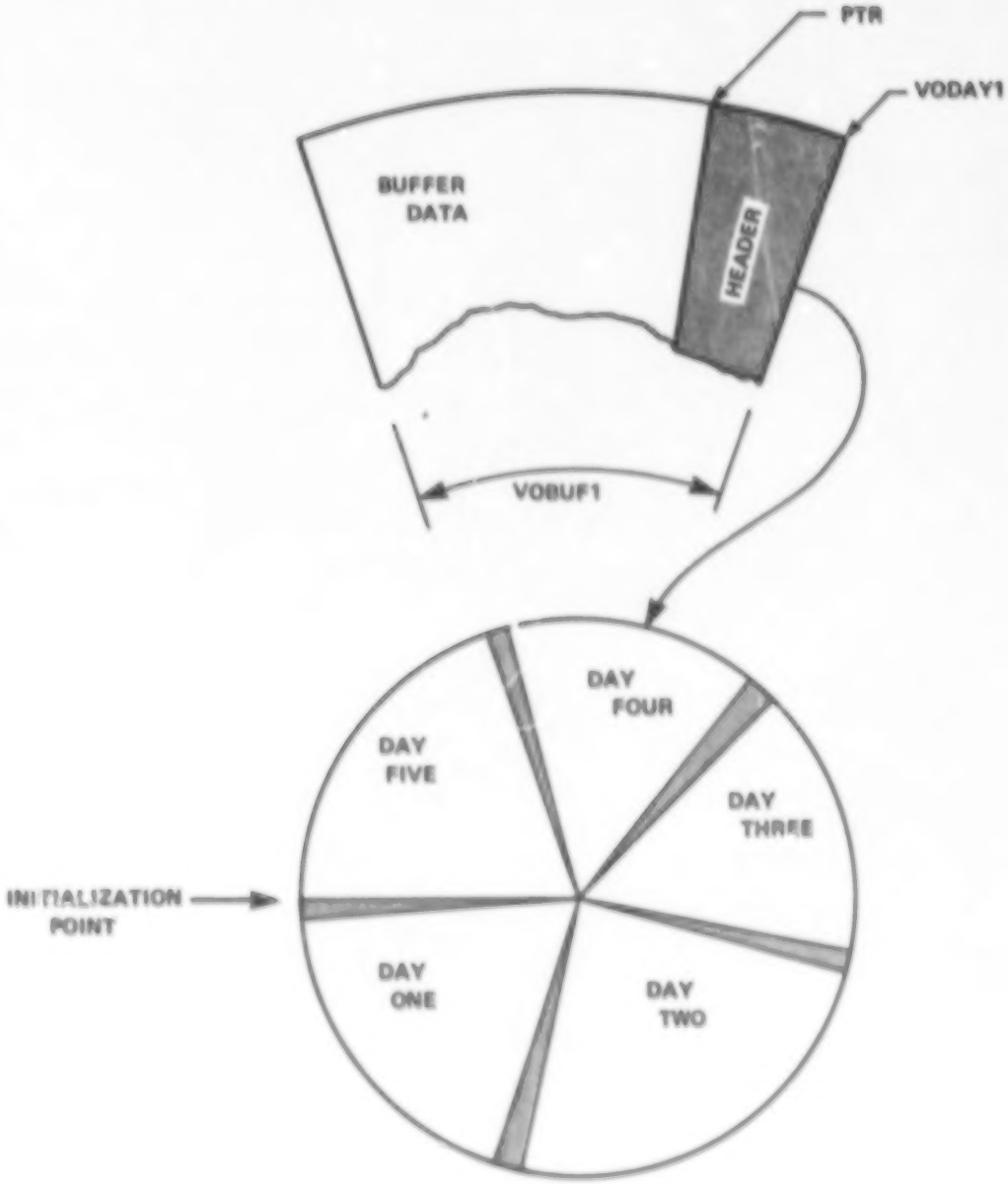


Figure 6.1/ Typical Five-Day Buffer Structure

6.8.3/ Parameter Index

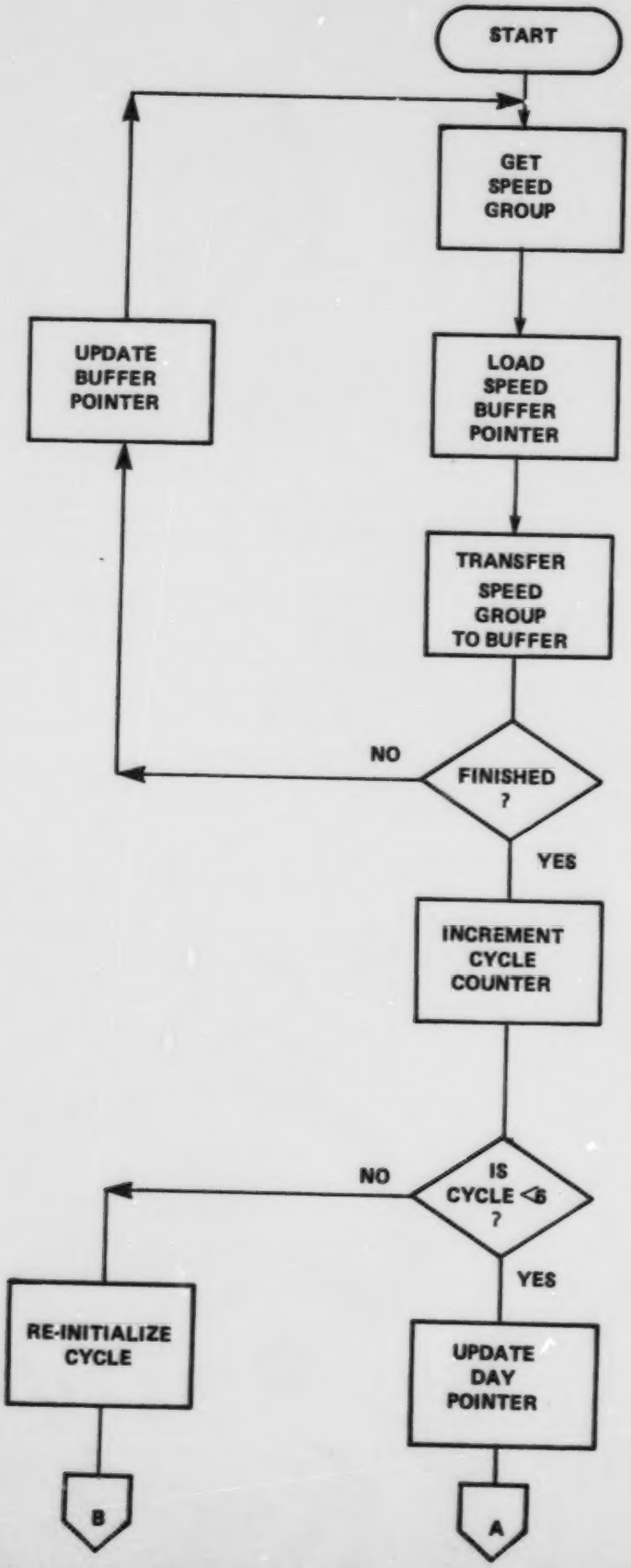
Routine: 'HEADER'

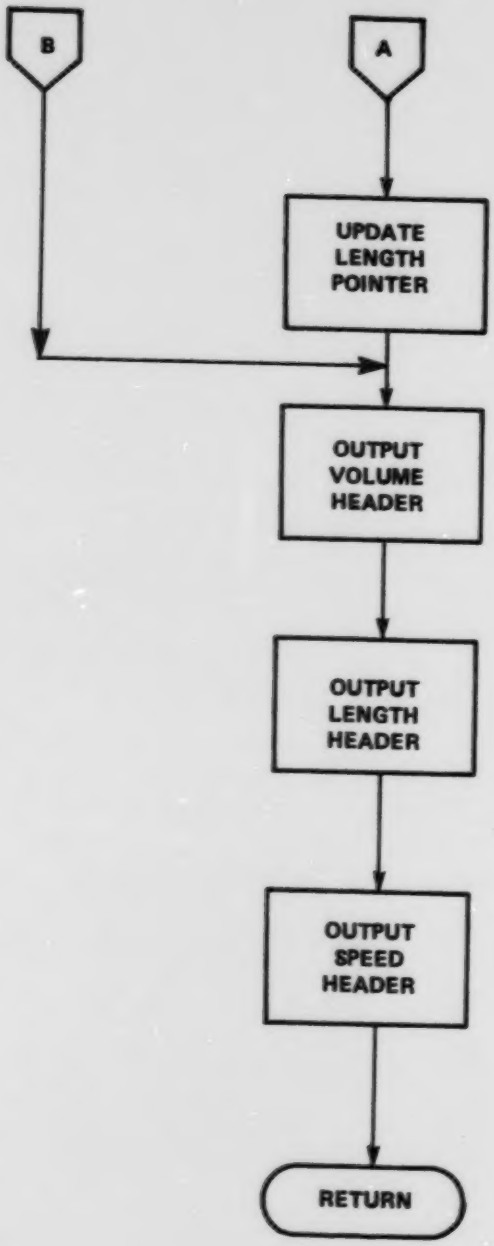
Description: Speed output and header control.

Parameter	Description
TEMPØ3	- a temporary storage area used in varied applications.
TEMPØ6	- a temporary storage area used in varied applications.
SPED	- a temporary base address for the classified daily speed data.
PTRS	- pointer to beginning of speed data.
TEMPØ7	- a temporary storage area used in varied applications.
CYCLE	- a parameter used to control the cycling of the circular buffers.
VOLDAY	- number of days volume data collected.
SPDDAY	- number of days of speed data collected.
LENDAY	- number of days of length data collected.
BEGIN	- label for destination of branch statement.

Parameter	Description
VODAY1	- pointer to beginning of day's volume buffer.
VOBUF1	- length of an individual day's volume buffer.
PTR	- pointer to beginning of volume data.
VODAYL	- pointer to beginning of current day's length buffer.
PTRL	- pointer to beginning of length data.
VODAYS	- pointer to beginning of current day's speed buffer.
START	- initial volume buffer pointer at Ø4ØØ.
LENGTH	- initial length buffer pointer.
SPEED	- initial speed buffer pointer at Ø3ØØ.
YEAR	- current calendar year (BCD).
MONTH	- current calendar month (BCD).
DAOFMO	- current day of the month (BCD).
DAOFWK	- current day of the week (BCD).

6.8.4/ Flow Chart




B
A
UPDATE LENGTH POINTER
OUTPUT VOLUME HEADER
OUTPUT LENGTH HEADER
OUTPUT SPEED HEADER
RETURN

6.9 'INPUT'

6.9.1/ Program Listing Reference

The relevant program listing may be found in Appendix A, starting on Page 171, Line 945, extending to Line 1Ø28, which includes both directions.

6.9.2/ Program Explanation

This, the 'INPUT' routine, is one of the most important operations of all the software, since without fast accurate detector sampling, the rest of the process is pointless. The first step is to set up the MSB of the port addresses, which is hexidecimal 5Ø. In PCS, the four ports used to accommodate 32 lanes are 5ØØ1, 5ØØ2, 5ØØ3 and 5ØØ6. Having defined part of the port address, the detector input areas and pointers are then cleared to ensure the integrity of the process. Next, the detector hardware assignment table is consulted.

The object of the table is to supply the input port number, the second byte to match 5Ø already supplied, and the bit position within the port for any and all the 32 detectors. The processing portion of the program can step through the table to the addressing details for any detector, plus acquire a reference byte for comparison to determine whether the detector is, in fact, occupied. In fact, that is how the main body of the program functions.

Once the pointer to the hardware assignment table has been established, and the port defined and accessed, the information is checked against that detector's reference byte. Negative results initiate a similar test of the next lane with reassigned pointers, while a positive result allows this detector to be added to the list of currently unverified detector inputs. The subject of data verification is treated in 'PCS2.' Next, as in the negative case, detector position bits are repositioned, tests are done to determine the status of the input sampling process and more detectors are sampled, if appropriate.

Upon completion of these tasks, the routine terminates and relinquishes control to the calling routine. This leaves the system with a list of detectors which are, and are not, occupied.

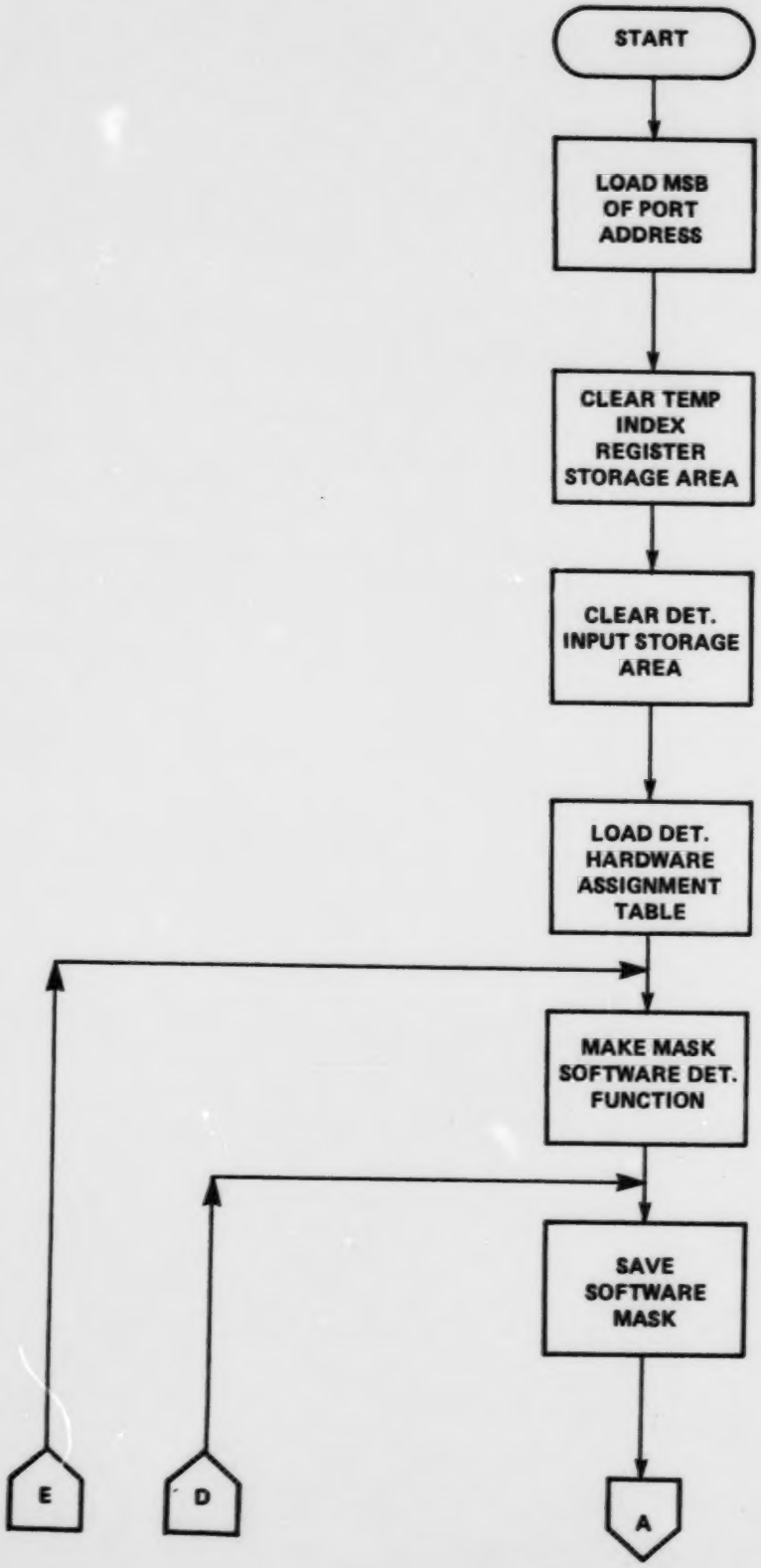
6.9.3/ Parameter Index

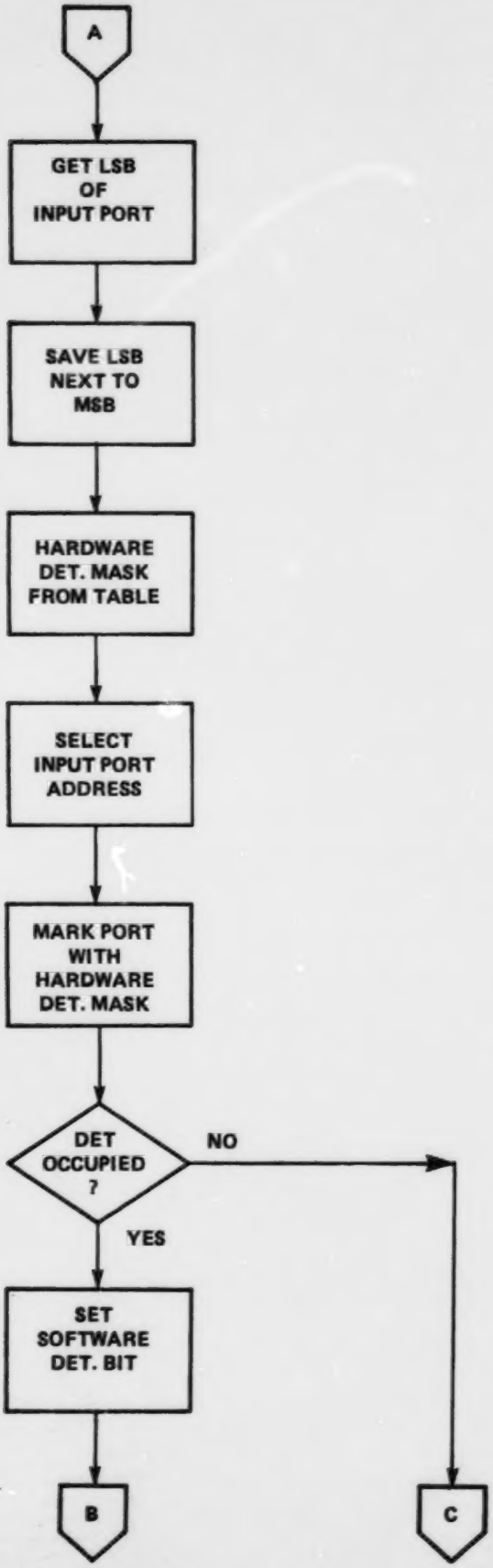
Routine: 'INPUT'

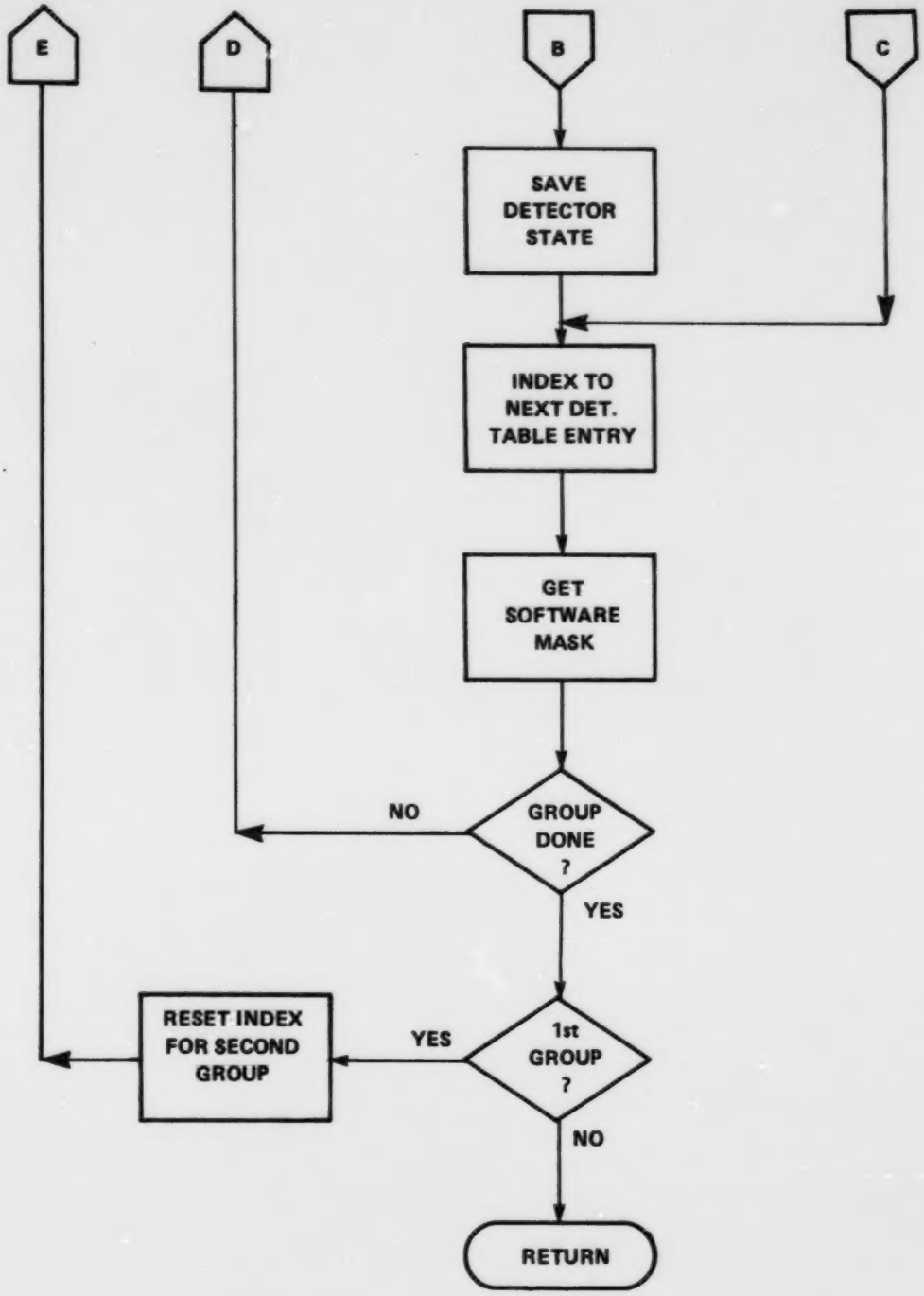
Description: Detector input scan control.

Parameter	Description
TEMPØ1	- a temporary storage area used for varied applications.
INDEX	- a variable used for the storage of the software detector index.
DATAØM	- MSB of current unverified detector data.
DTRTBL	- label for detector hardware table.
TEMPØ4	- a temporary storage area used for varied applications.
TEMPØ2	- a temporary storage area used for varied applications.
NXTDET	- branch destination for next lane consideration.
DTCTRM	- MSB of detector enable mask.
IPT2	- label for branch destination.
DONE	- label for branch destination.
IPT1	- label for branch destination.

6.9.4/ Flow Chart




A
GET LSB OF INPUT PORT
SAVE LSB NEXT TO MSB
HARDWARE DET. MASK FROM TABLE
SELECT INPUT PORT ADDRESS
MARK PORT WITH HARDWARE DET. MASK
DET OCCUPIED ?
NO
YES
SET SOFTWARE DET. BIT
B
C


E
D
B
C
SAVE DETECTOR STATE
INDEX TO NEXT DET. TABLE ENTRY
GET SOFTWARE MASK
GROUP DONE ?
NO
YES
1st GROUP ?
YES
NO
RESET INDEX FOR SECOND GROUP
RETURN

6.10/ 'PCS2'

6.10.1/ Program Listing Reference

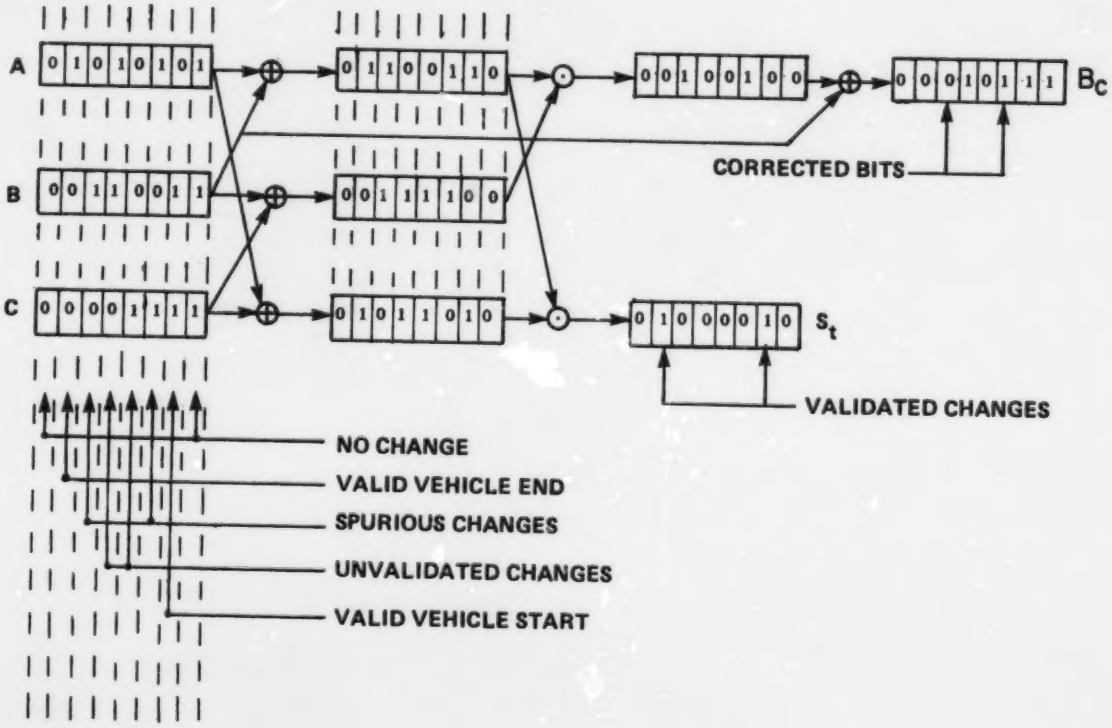
The relevant program listing may be found in Appendix A, starting on Page 183, Line 1911, extending to Line 1962, which includes both directions.

6.10.2/ Program Explanation

This single error correction is a logical process which discriminates all incoming signals from both 'INPUT' and 'INPUTT.' The purpose of the sequence is to ensure that spurious data detector changes don't filter through as valid data. For example, a valid vehicle start is represented by a sequence of three data samples, such as low, high, high. In the same manner, a valid vehicle end is represented by high, low, low. Any other sequence represents either no or an invalid change (Figure 6.2). As a result of this process, the unverified data provided by the input routines are either accepted or disqualified by the verification routine. If accepted, the appropriate state change bits are set to form a very effective technique for eliminating erroneous input data.

DATA ϕ = C CURRENT DETECTOR DATA (DATA ϕM, DATAϕL)
DATA 1 = B NEXT MOST CURRENT DETECTOR DATA (DATA1M, DATA1L)
DATA 2 = A 2nd FROM CURRENT DETECTOR DATA (DETINM, DETINL)

VERIFIED DETECTOR DATA = $B_c = (A \oplus B \cdot B \oplus C) \oplus B$
STATE CHANGE DATA = $S_t = (A \oplus C \cdot A \oplus B)$



$\odot$ = AND

$\oplus$ = EXCLUSIVE OR

Figure 6.2/ Single Error Correction

6.10.3/ Parameter Index

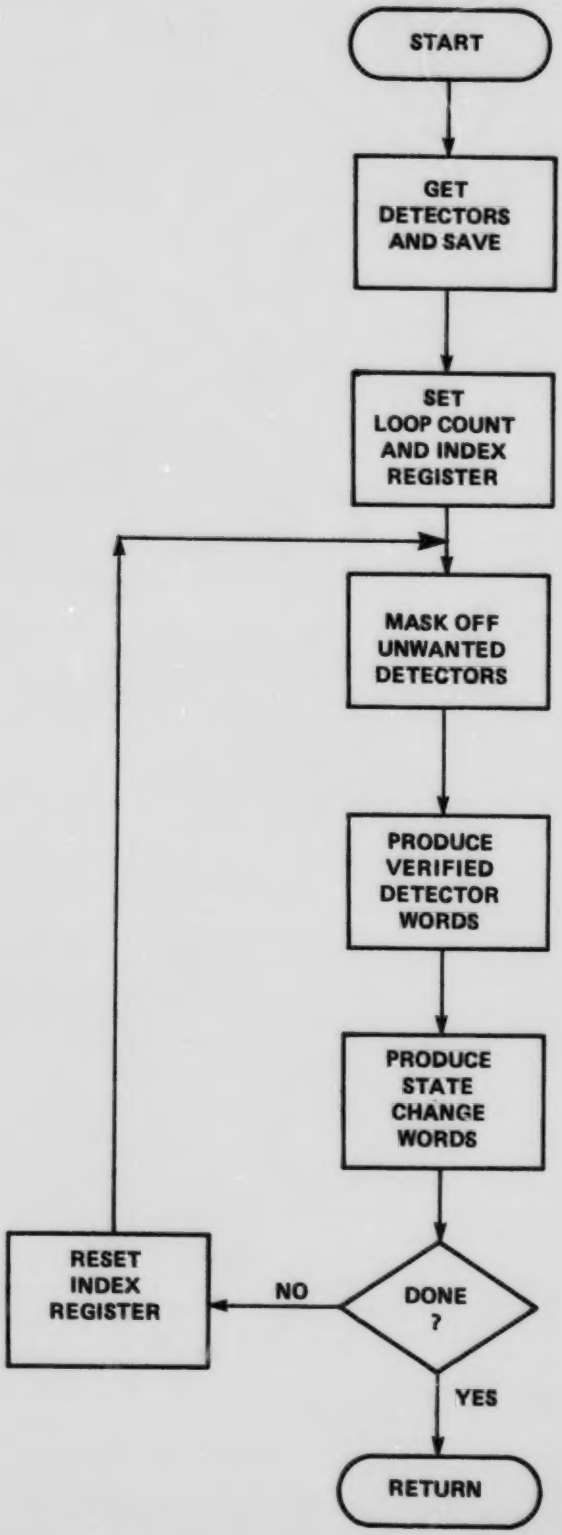
Routine: 'PCS2'

Description: Single error correction routine.

Parameter	Description
INPUT	- label for detector sampling routine in Direction 1.
INPUTT	- label for detector sampling routine in Direction 2.
SØ	- label for branch destination.
TEMPØ4	- a temporary storage area used for varied applications.
TEMPØ5	- a temporary storage area used for varied applications.
DATAØL	- LSB of current unverified detector data.
TEMPØ6	- a temporary storage area used for varied applications.
DATAØM	- MSB of current unverified detector data.
DTCTRM	- MSB of detector enable mask.
TEMPØ7	- a temporary storage area used for varied applications.

Parameter	Description
TEMPØ5	- a temporary storage area used for varied applications.
S1	- label for branch destination.

6.10.4/ Flow Chart



6.11/ 'PCS3'

6.11.1/ Program Listing Reference

The relevant program listing may be found in Appendix A, starting on Page 185, Line 1989, extending to Line 2135, which includes both directions.

6.11.2/ Program Explanation

This routine provides the initial service to the detector inputs. It adds counts for the occupancy, volume and the speed loop transition timers. In the case of volume and its associated occupancy, it undergoes only data manipulation from this point, whereas speed still requires several input values.

The process is introduced by incrementing two counters or timers, depending upon your point of view. These counters record the number of 1/60 of a second counts experienced between the up and downstream loops for each enabled lane. The exercise is actually done twice, since notice is given to vehicles both entering and leaving each detector. The transition timing flags are kept high for any lane which is eligible to be incremented and dropped low when a vehicle traverses the downstream detectors.

The next operation is to see if the upstream detector is occupied. This proving to be true, the routine increments the 6 s occupancy and the long vehicle occupancy before updating the transition timing flags. On the other hand, if it proves to be false, that is, the particular lane in question is not occupied, then the state change bit is examined. Here, there are two possibilities: either a vehicle could have left the detector area setting the state change bit (see 'PCS2'), or there may not have been any vehicles present at all. In the latter case, the process is simply started over again for the next lane in the sequence.

If there has been a state change, the 6 s volume count is incremented to indicate the passage of the vehicle. Then the state change bit is reset, followed by the updating of the transition timer flag and the detector error bit. At this point, the program rejoins the other alternatives and selects the next lane for processing by the use of the subroutine 'NEXTLN.' Upon completion of all the lanes, the control is returned to the calling routine.

6.11.3/ Parameter Index

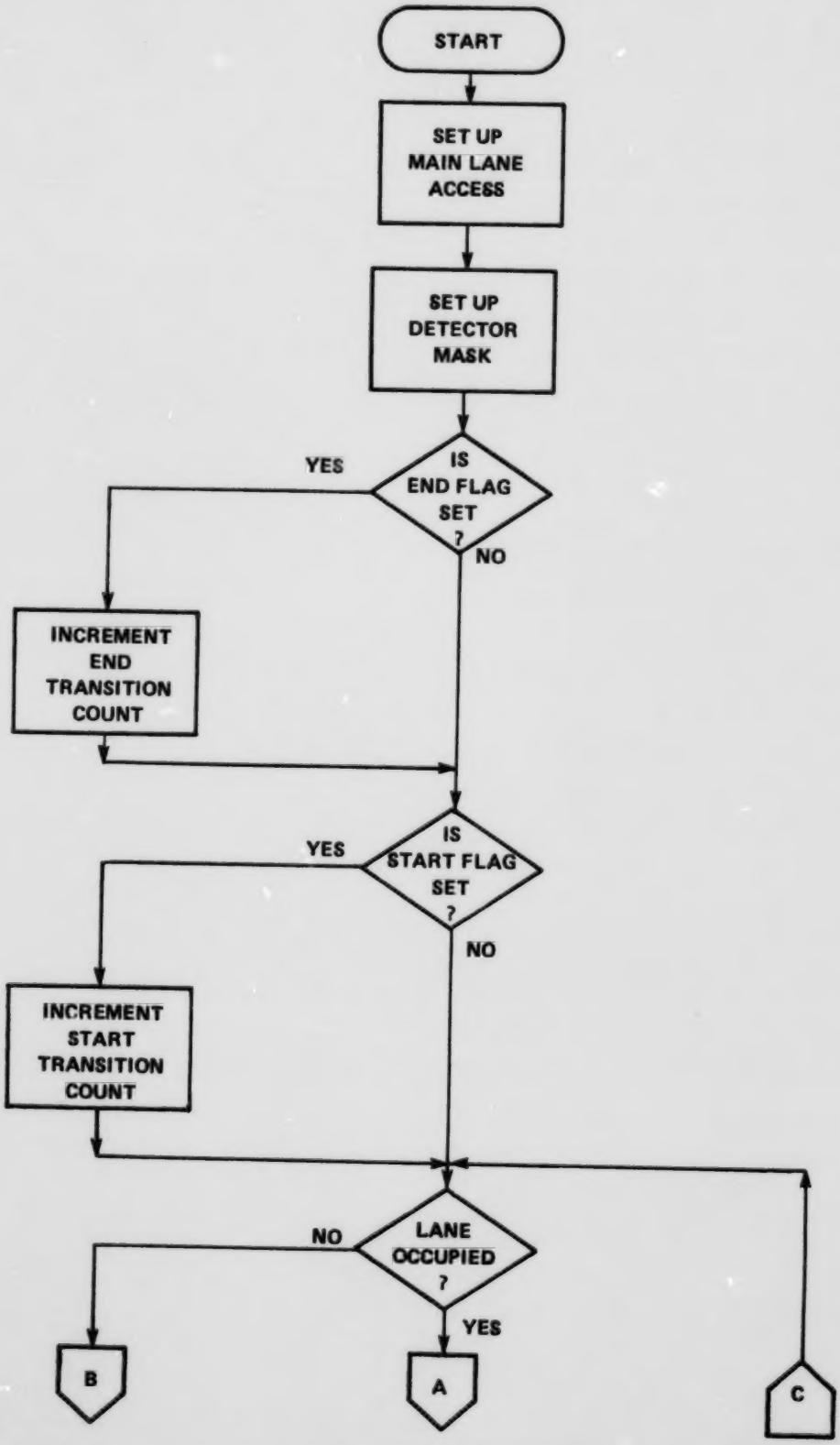
Routine: 'PCS3'

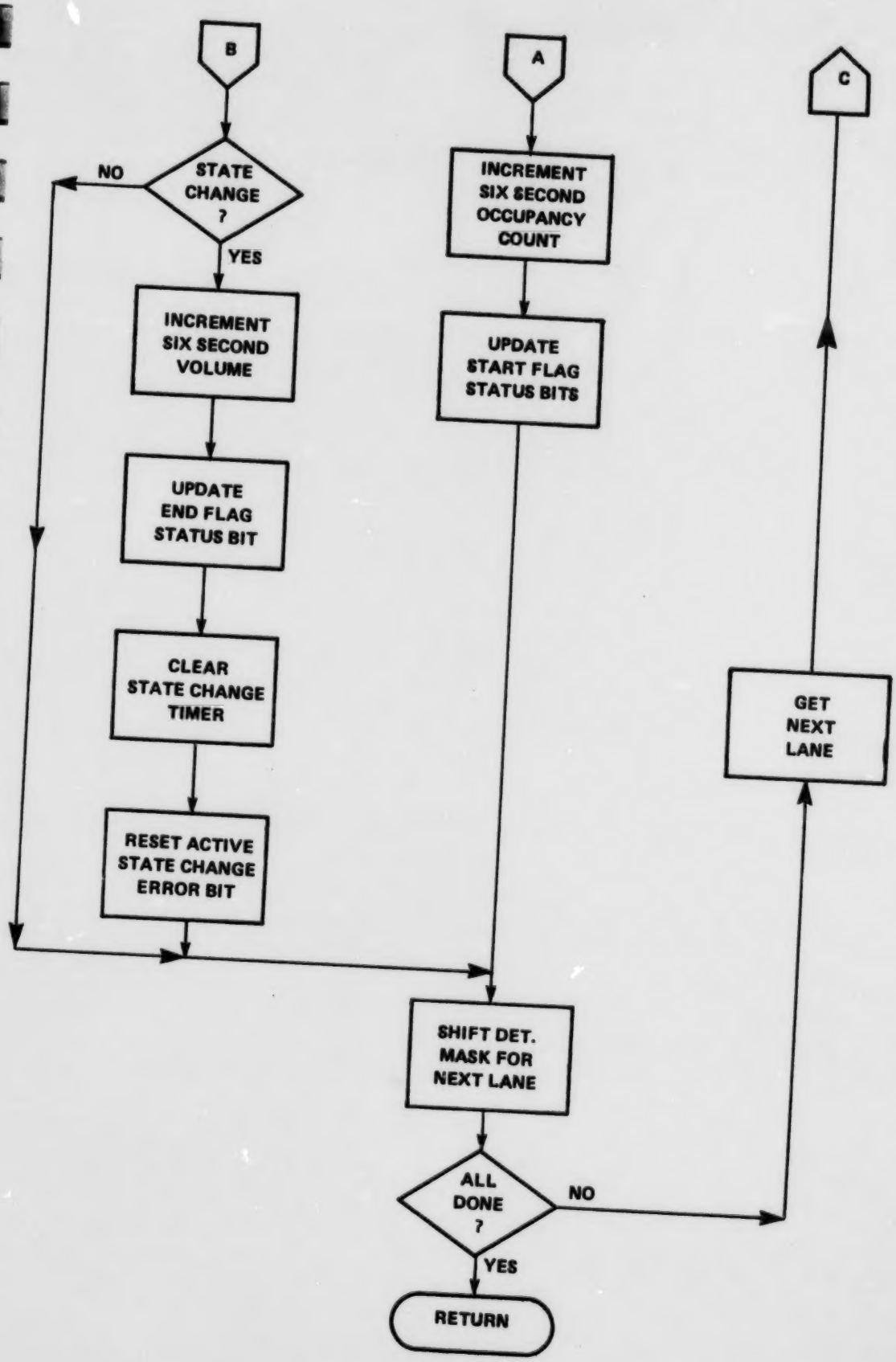
Description: Upstream detector servicing routine.

Parameter	Description
SETUP	- subroutine call to initialize lane data pointer.
TBIT7	- this parameter represents the predefined value of $8Ø.
TEMPØ3	- a temporary storage area used for varied applications.
TEMPØ1	- a temporary storage area used for varied applications.
GO	- Direction 1 vehicle end transition timing flag.
T20	- label for internal branch statement.
B1E	- vehicle end transition timer.
GOT	- Direction 1 vehicle start transition timing flag.
T21	- label for internal branch statement.
B1S	- vehicle start transition timer.
DETINM	- MSB of verified detector data.

Parameter	Description
T16	- label for internal branch statement.
STCHGM	- MSB of detector state change word.
T14	- label for internal branch statement.
VSIX1	- parameter holds the lane's 6 s count.
STCHG1	- state change timer.
DET1	- a byte containing the lane's detector error flags.
OSIX1	- parameter holds the lane's 6 s occupancy count.
OCC	- variable that contains the loop's occupancy by vehicle.
NEXTLN	- subroutine call to invoke the lane scan control routine.
T1	- label for internal branch statement.

6.11.4/ Flow Chart




B
A
C
STATE CHANGE ?
NO
YES
INCREMENT SIX SECOND VOLUME
UPDATE END FLAG STATUS BIT
CLEAR STATE CHANGE TIMER
RESET ACTIVE STATE CHANGE ERROR BIT
INCREMENT SIX SECOND OCCUPANCY COUNT
UPDATE START FLAG STATUS BITS
GET NEXT LANE
SHIFT DET. MASK FOR NEXT LANE
ALL DONE ?
NO
YES
RETURN

6.12/ 'PCS3B'

6.12.1/ Program Listing Reference

The relevant program listing may be found in Appendix A, starting on Page 188, Line 2160, extending to Line 2246, which includes both directions.

6.12.2/ Program Explanation

The purpose of the downstream detectors is to simply mark the end of a vehicle's excursion on the test section. In terms of the software, the routine must update the transition timing flags so as to allow proper termination of vehicle transition timers. (See 'PCS3.')

The program starts by setting the lane pointer via a subroutine call, then loading the lane detector mask for the highest number lane. Then the verified detector data are compared against the mask to determine whether or not a vehicle is present. If the test is true, a vehicle is present; then the vehicle start transition flag is dropped, disallowing any further counts for its respective counter found in 'PCS3.' On the other hand, if no vehicle is present, a further test for state change must be conducted. If the state has not changed, then the program concludes that there wasn't any action necessary for this lane and goes on to the next. But where the state change indicates a vehicle has just left, the vehicle end transition flag is dropped, disallowing any further counts in its respective counter. Either way, the program steps through the lanes until completion, whereby control is relinquished to the calling routine.

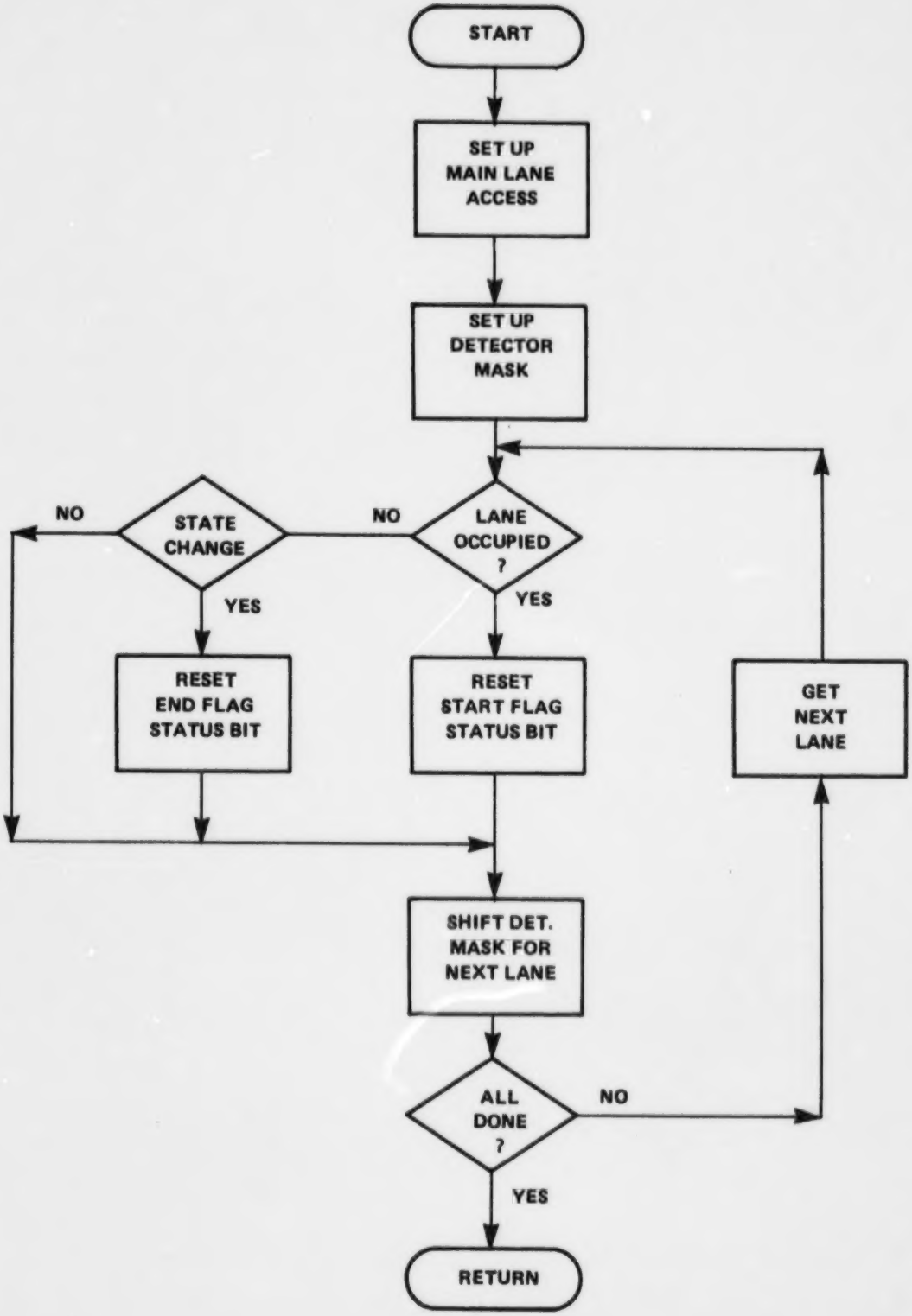
6.12.3/ Parameter Index

Routine: 'PCS3b'

Description: Downstream detector servicing routine.

Parameter	Description
SETUP	- subroutine call to initialize lane data pointer.
TBIT7	- the parameter represents the predefined value of $8Ø.
DETINM	- MSB of verified detector data offset by $2Ø for downstream.
TB16	- label for internal branch statement.
STCHGM	- MSB of detector state change word offset by $2Ø for downstream.
TB14	- label for internal branch statement.
GO	- Direction 1 vehicle end transition timing flag.
GOT	- Direction 1 vehicle start transition timing flag.
NEXTLN	- subroutine call to invoke the lane scan control routine.
TB1	- label for internal branch statement.

6.12.4/ Flow Chart



6.13/ 'PCS4'

6.13.1/ Program Listing Reference

The relevant program listing may be found in Appendix A, starting on Page 172, Line 1Ø61, extending to Line 124Ø, which includes both directions.

6.13.2/ Program Explanation

The purpose of this routine is to use the 6 s volume/occupancy counts to calculate hourly volume and the running totals for volume/occupancy. The accumulation of the hourly volume count is very straightforward, while the running totals are somewhat more involved.

The routine starts, after setting the lane index pointer, by examining the data clear flag 'KAY1.' If this indicator is non-zero, the lane's data area is cleared of its present values. In a routine which, in part, calculates pseudo running totals, this is of obvious importance. As can be seen in Section 6.6.2, proportioning is done by taking 9/10 of a volume and adding the current 6 s count. Therefore, residue in data area to start with is undesirable. It is not necessary to dwell on the process, since this running total feature is available but not utilized by a standard PCS station.

The remainder of the routine manipulates data in preparation for proportioning, as well as accumulating the value count into the desired hourly format. Information from here goes into the 'PCS5' series of routines to further process information into average running totals.

6.13.3/ Parameter Index

Routine: 'PCS4'

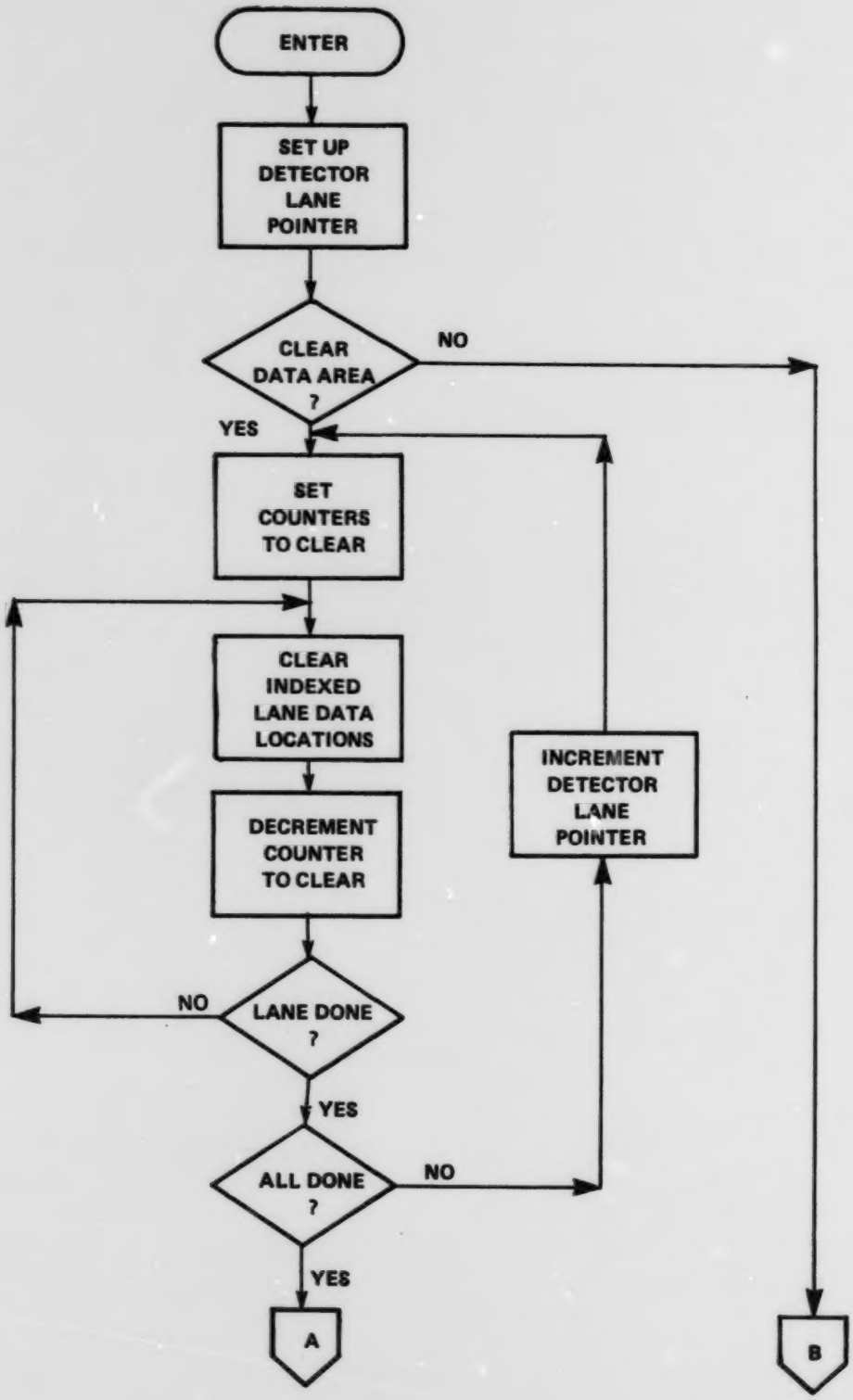
Description: Lane data handling routine.

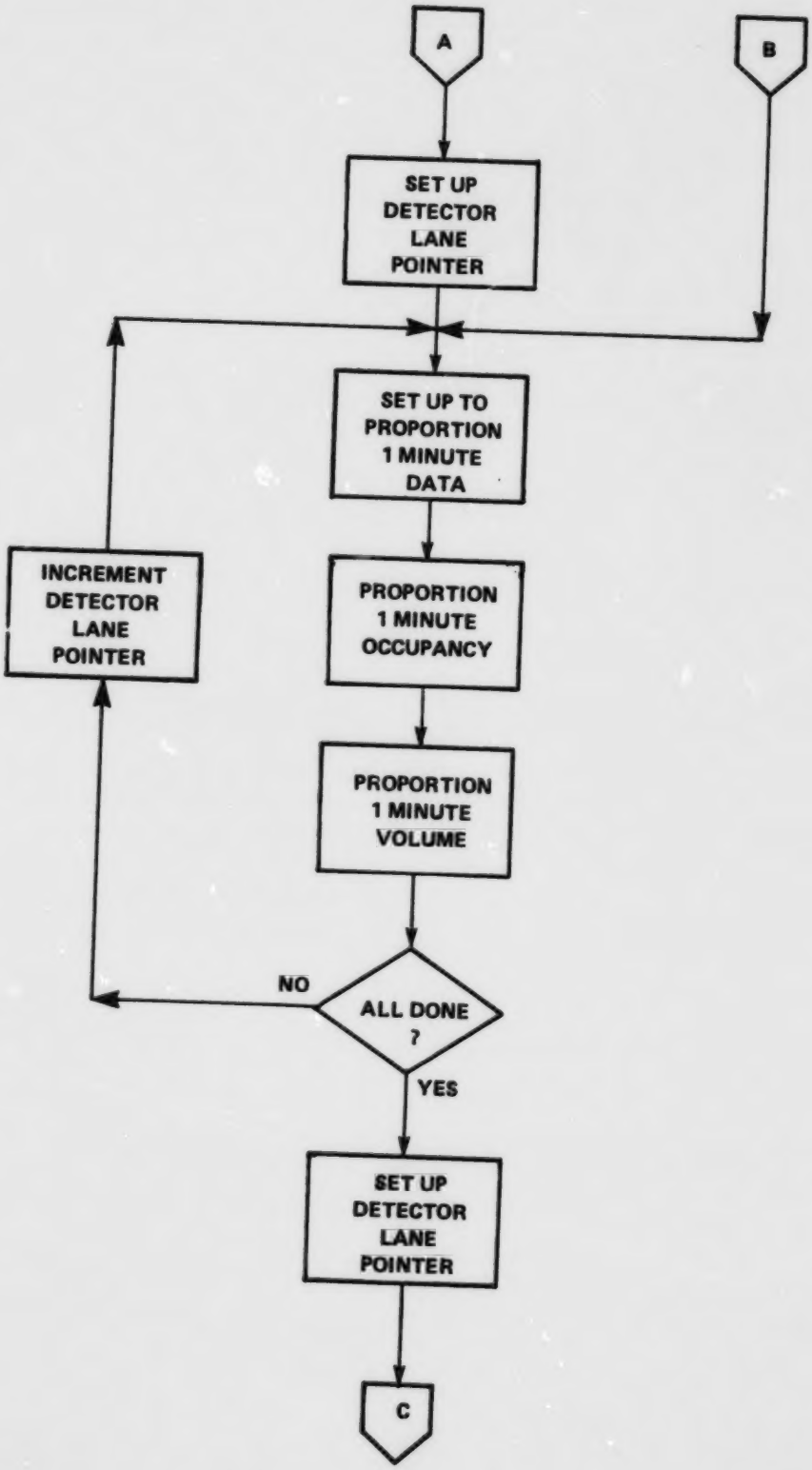
Parameter	Description
SETUP	- subroutine call to initialize lane data pointer.
TEMPØ4	- a temporary storage area used for varied applications.
KAY1	- clear flag for detector data area.
PROP1	- label for internal branch statement referencing 1 min average running total proportioning.
OSIX1	- byte containing 6 s occupancy.
D2	- label for internal branch statement.
NEXTLN	- subroutine call to advance lane data pointer.
D1	- label for internal branch statement.
PCW1	- proportion control word, set to '1A' (26/256) for 1 min.
OCMN1M	- MSB of occupancy for 1 min running total.

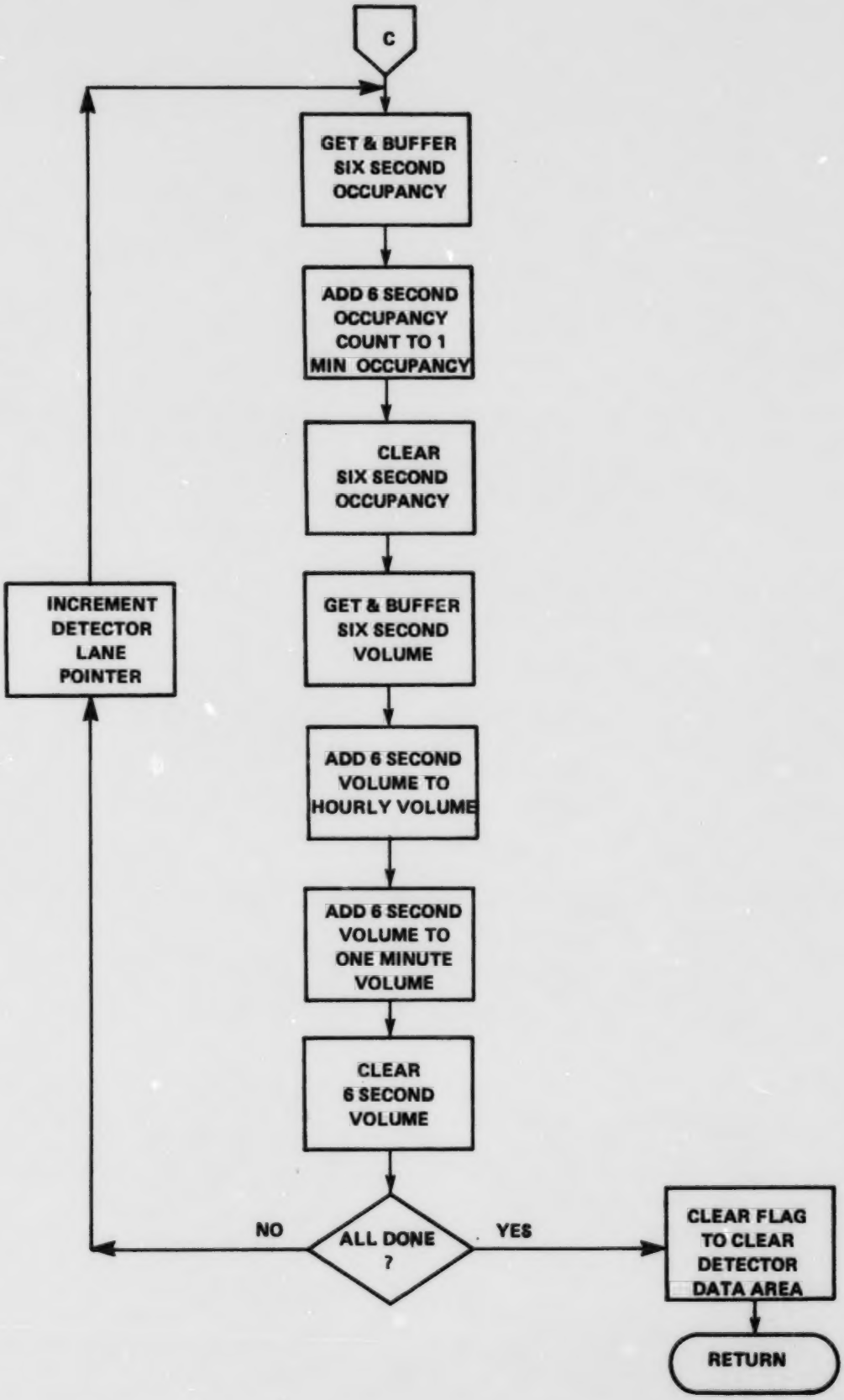
Parameter	Description
PRO16	- label for branch statement destination to set up for 2-byte proportioning.
VMN1	- 1 min running total volume.
PRO8	- label for branch statement destination to set up for 1-byte proportioning.
NEXTLN	- subroutine call to invoke the lane scan control routine.
OCC1L	- LSB of an occupancy count.
D3	- label for internal branch statement destination.
OCC1M	- MSB of an occupancy count.
OCMN1L	- LSB of occupancy for 1 min running total.
D5	- label for internal branch statement destination.
VSIX1	- byte containing 6 s volume count.
VOL1	- volume count.
VD1L	- LSB of the hourly accumulated volume count (cleared hourly).
D56	- label for internal branch statement destination.

Parameter	Description
VD1M	- MSB of the hourly accumulated volume count (cleared hourly).
D50	- label for internal branch statement destination.
INDEX	- stores values of the index register.
TEMPØ5	- a temporary storage area used for varied applications.
TEMPØ4	- a temporary storage area used for varied applications.
PRPRT8	- subroutine call to invoke proportioning for pseudo running totals.

6.13.4/ Flow Chart




A
B
SET UP DETECTOR LANE POINTER
SET UP TO PROPORTION 1 MINUTE DATA
PROPORTION 1 MINUTE OCCUPANCY
PROPORTION 1 MINUTE VOLUME
ALL DONE ?
NO
YES
INCREMENT DETECTOR LANE POINTER
SET UP DETECTOR LANE POINTER
C


C
GET & BUFFER SIX SECOND OCCUPANCY
ADD 6 SECOND OCCUPANCY COUNT TO 1 MIN OCCUPANCY
CLEAR SIX SECOND OCCUPANCY
GET & BUFFER SIX SECOND VOLUME
ADD 6 SECOND VOLUME TO HOURLY VOLUME
ADD 6 SECOND VOLUME TO ONE MINUTE VOLUME
CLEAR 6 SECOND VOLUME
ALL DONE ?
NO
YES
INCREMENT DETECTOR LANE POINTER
CLEAR FLAG TO CLEAR DETECTOR DATA AREA
RETURN

6.14/ 'PCS5'

6.14.1/ Program Listing Reference

The relevant program listing may be found in Appendix A, starting on Page 174, Line 127Ø, extending to Line 1455, which includes both directions.

6.14.2/ Program Explanation

Although the running and average totals are not actively used now, the potential in the future for control function could prove invaluable.

The program starts by clearing all the old average running totals. Then it steps through all the lanes to determine which are eligible for consideration by discriminating the detector error flag 'DET1,' set by the 'PCS3' series of routines. Next the eligible lanes are checked to see if they have been enabled, 'INIT' routine, for use in this particular installation before proceeding.

Once the lanes have been decided upon, the running totals for both volume and occupancy update the average running total table (ART). This is done for all lanes involved. Upon completion, the remainder of the program is concerned with proportioning the summed running totals based on the number of lanes. This process relies on the proportioning subroutine 'PRPRT' and the predefined lane proportioning constants. An explanation of the latter may be seen in Section 2.4. As a result of 'PCS4' and 'PCS5,' there are running and average running totals for both volume and occupancy. These features are not utilized in this phase of PCS, but are available if any control operations require them.

6.14.3/ Parameter Index

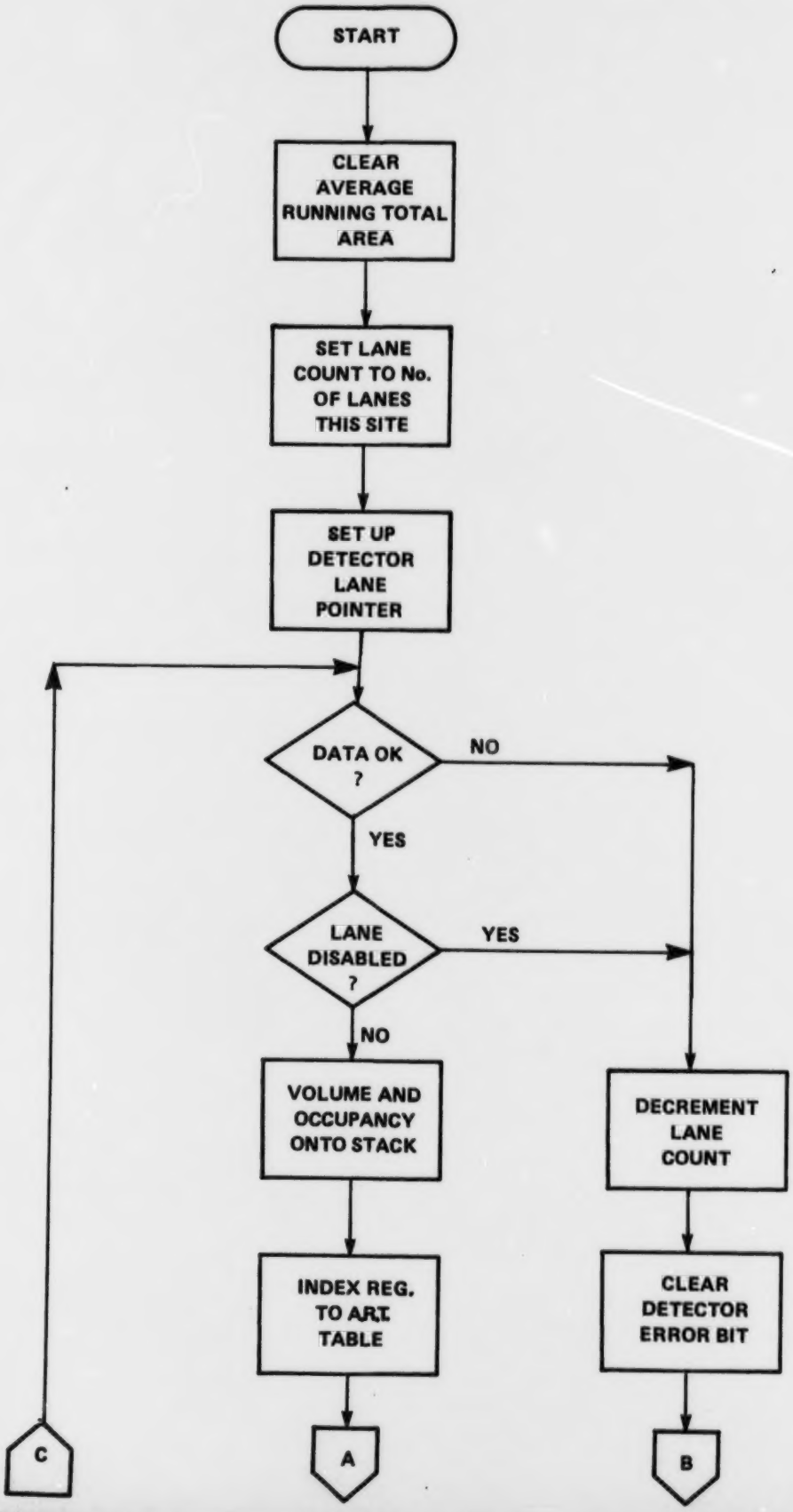
Routine: 'PCS5'

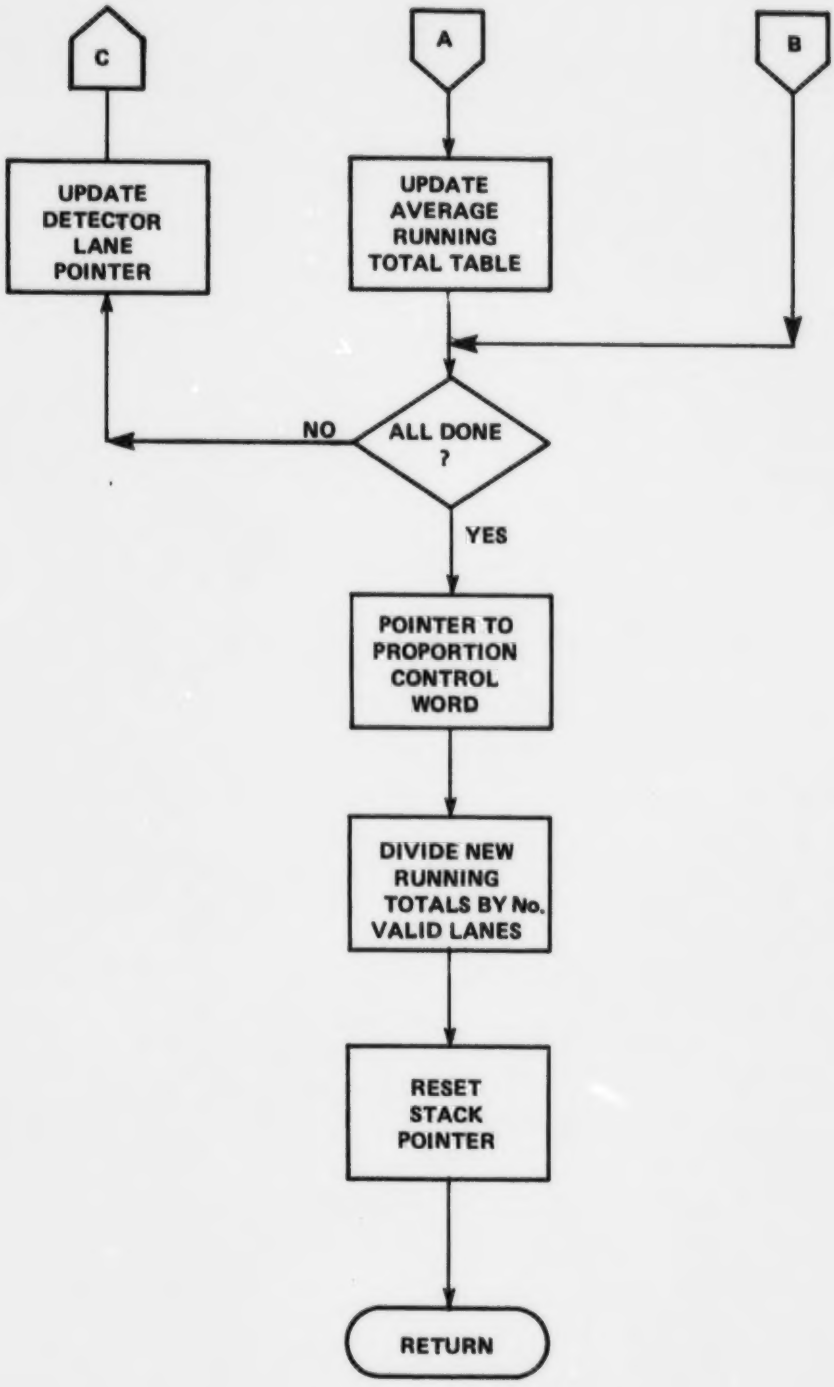
Description: Average running total routine.

Parameter	Description
AR1MOM	- MSB of average running 1 min occupancy.
LOOP1	- label for internal branch statement.
SETUP	- subroutine call to initialize lane data pointer.
DET1	- detector error flag.
E5	- label for internal branch statement.
NLOOP2	- label for internal branch statement.
LNDIS1	- lane disable flag. (Ø allows lane data to be processed.)
E3	- label for internal branch statement.
VMN1	- 1 min volume running total.
PRPRT	- a subroutine call enabling the proportioning routine for pseudo running total (Section 6.6).

Parameter	Description
OCMN1M	- MSB of 1 min occupancy running total.
OCMN1L	- LSB of 1 min occupancy running total.
NEXTLN	- subroutine call to invoke the lane scan control routine.
LOOP2	- label for internal branch statement.
PCWTAB	- label for internal branch statement to proportion control word table in routine.
PIKUP	- label for internal branch statement.
LOOP4	- label for internal branch statement.
INDEX	- temporary storage for index register representing, in this case, ART table pointer.
AR1MVL	- LSB of average running 1 min volumes.
OUT	- label for internal branch statement.
LOOP5	- label for internal branch statement.
PCLN1-8	- predefined lane proportioning constants.

6.14.4/ Flow Chart




C
A
B
UPDATE DETECTOR LANE POINTER
UPDATE AVERAGE RUNNING TOTAL TABLE
NO
ALL DONE ?
YES
POINTER TO PROPORTION CONTROL WORD
DIVIDE NEW RUNNING TOTALS BY No. VALID LANES
RESET STACK POINTER
RETURN

6.15/ 'PCS6'

6.15.1/ Program Listing Reference

The relevant program listing may be found in Appendix A, starting on Page 178, Line 1484, extending to Line 1628, which includes both directions.

6.15.2/ Program Explanation

The whole purpose of this routine is to monitor, in a minor way, the lane's detector status. Although only one type of error is checked for, several more will inevitably prove desirable as time goes on. With this in mind, Bits 1 to 7 are reserved for individual failures, while Bit 8 is set if there is a failure of any nature. The software is designed to check for suspiciously long occupancies by comparing the 1 min values with a preset limit.

The routine is introduced by the definition of two error parameters, '01BIT' and 'ERRBIT,' to take the values $82 and $8Ø, respectively. Next, there is an imposing statement which, if pondered, reveals that the value of the 1 min occupancy inhabits $Ø2Ø1. With this established, the occupancy limit, established in 'INIT,' is compared to the 1 min occupancy with the use of 'CMP16' found in Section 5.5. In the case where there is no violation of the limit, the routine tries the next lane. On the other hand, if occupancy is too large, a detector error is flagged. The #01BIT value is superimposed on 'DET1' if the lane has been enabled. The only area to be completed is the error flags for external use.

So far the description has been limited to the process of setting 'DET1,' which is a variable for internal use. This variable is set and reset to reflect on continuing checks of the detectors. This is all well and good, but there is a need to know whether a failure has taken place which would affect the integrity of the data transfer to the transmit buffer. For this purpose, the rest of the routine sets the 'DETER1' byte, which logs any detector errors and can only be reset by external sources. Therefore, once the 'INIT' routine has reset 'DETER1,' it can be monitored anytime to determine the status of the lane's loop detectors. Where any failure will be indicated by setting the appropriate bit, it will be held until the system is either reset or manually changed. With this complete, control is returned to the calling routine.

6.15.3/ Parameter Index

Routine: 'PCS6'

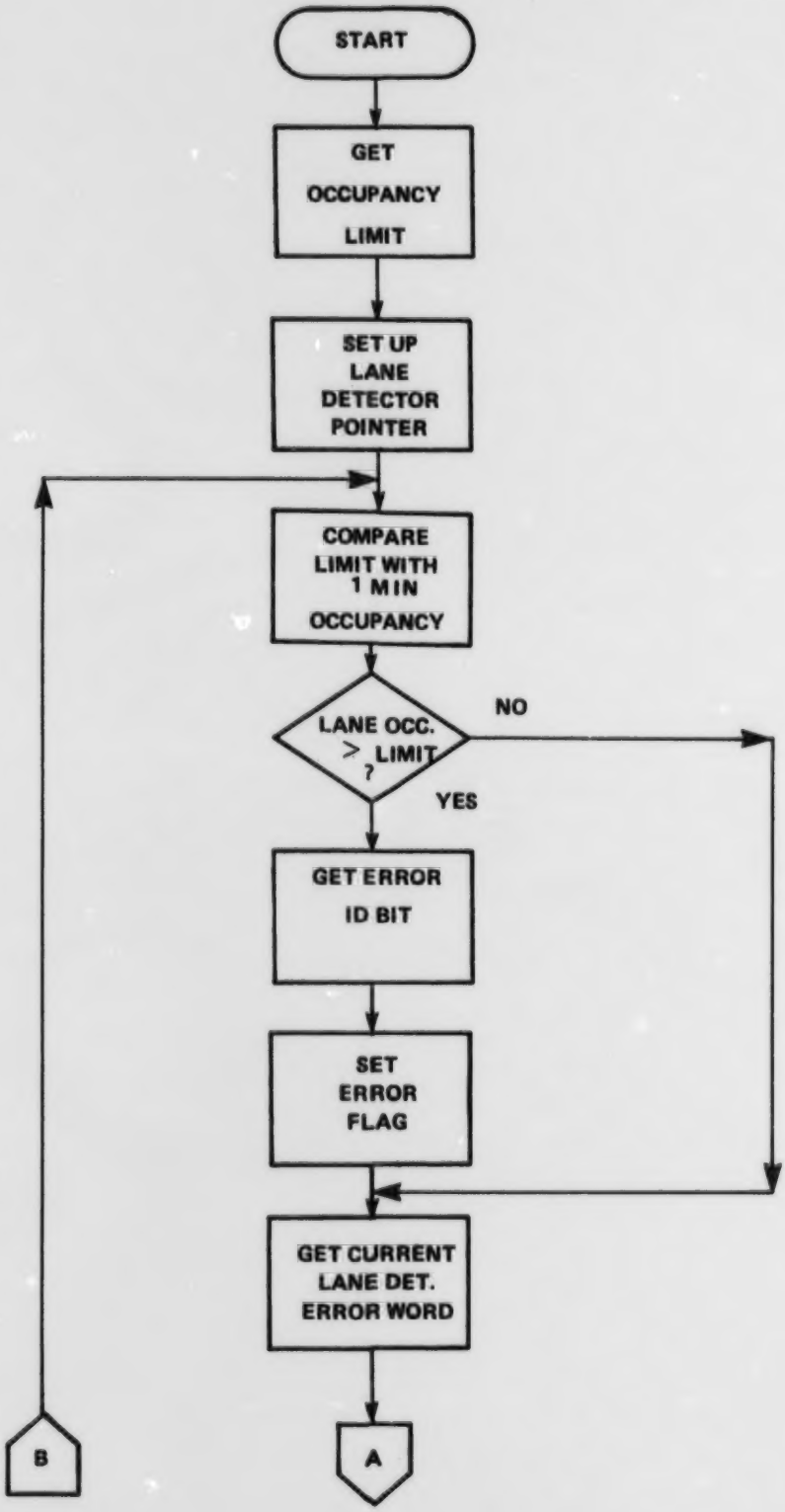
Description: Detector failure check routine.

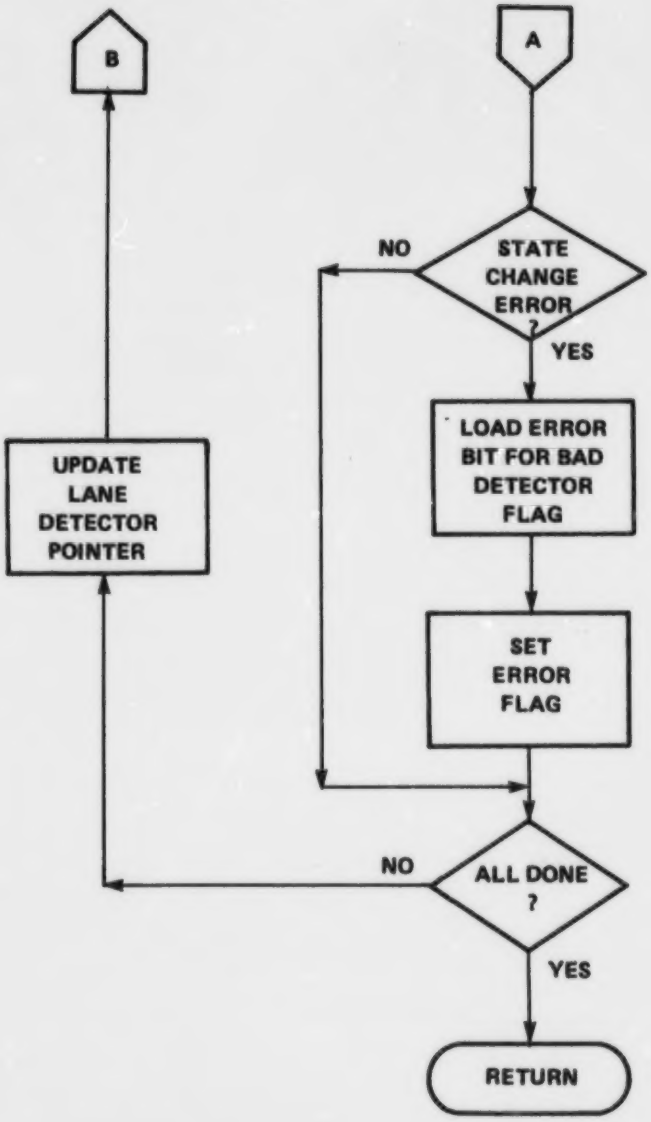
Parameter	Description
O1BIT	- a parameter defined to have the value $82.
ERRBIT	- a parameter defined to have the value $8Ø.
OCMN1M	- MSB of 1 min occupancy running total.
TEMPØ4	- a temporary storage area used for varied applications.
OC1OCM	- 1 min occupancy limit.
TEMPØ2	- a temporary storage area used for varied applications.
SETUP	- subroutine call to initialize lane data pointer.
OCTEST	- label for internal branch statement which is set up for 16-bit comparison.
LO7	- label for internal branch statement.
LO1	- label for internal branch statement.

Parameter	Description
ERSET	- label for internal branch to set detector error flag 'DET1.'
NEXTLN	- subroutine call to invoke the lane scan control routine.
LO11	- label for internal branch statement.
DETER1	- eight-lane detector error byte.
TEMPØ3	- a temporary variable used for varied applications.
DET1	- detector error byte.
LO13	- label for internal branch statement.
TEMPØ1	- a temporary variable used for varied applications.
LO12	- label for internal branch statement.
LNDIS1	- lane disable byte. (A non-Ø value indicates the lane is unused.)
ER9	- label for internal branch statement.
INDEX	- a variable used for storing the index register. Also uses 'INDEX+1.'

Parameter	Description
TEMPØ6	- a temporary storage area used for varied applications.
TEMPØ5	- a temporary storage area used for varied applications.
CMP16	- call for the generalized subroutine to compare two 16 bit numbers.

6.15.4/ Flow Chart




A
B
STATE CHANGE ERROR ?
NO
YES
LOAD ERROR BIT FOR BAD DETECTOR FLAG
SET ERROR FLAG
UPDATE LANE DETECTOR POINTER
ALL DONE ?
NO
YES
RETURN

6.16/ 'PCS7'

6.16.1/ Program Listing Reference

The relevant program listing may be found in Appendix A, starting on Page 180, Line 1653, extending to Line 1768, which includes both directions. See Page 190, Line 2270 to Line 2386 for 'PCS7A.'

6.16.2/ Program Explanation

This routine provides the transfer of the hourly volume data into the transmit buffer according to the configuration defined in initialization. To accomplish this, the routine first gets the current hour, which is in BCD, and converts it to binary. Then it locates the right location in the buffer for the given time. To do this, the start of the volume buffer is loaded and, if it happens to be the end of hour zero, this is taken as the correct buffer location. For any other hour, the value of 'SIZE' is continuously added to the start pointer 'PTR,' while the hour value is decremented until the correct location is reached. From this location onwards for a distance of 'SIZE,' the memory is zeroed to remove all traces of volume information for the time slot which is about to be used. The rest of the routine, which is the majority, deals with the transfer of data.

The process of summing and categorizing data is basically under the control of two types of variables, 'VOLW1' and 'NVOL1,' for each direction. The routine just steps through the eligible lane data, first noting the number of volume counts to be assessed. For each count, to be assessed, it then consults in sequence the various lane configurations to be summed for the particular count. Along with clearing the hourly volume count from the locations from which they are extracted, the process is allowed to work until completion, where it returns control, once again, to the calling routine. A printout of a typical day's hourly volume counts obtained by telemetry link can be seen in Figure 6.3.

1. PCS STATION I.D. : CONTROLLER #21, STATION #52 -- LAKESHORE
2. HIGHWAY : 2 (EAST OF CREDIT RIVER BRIDGE)
3. DATE : 82/ 9/14, TUESDAY

4. VOLUME COUNT :

HOUR	WESTBOUND (2 LANES)	EASTBOUND (2 LANES)
1	135.	88.
2	86.	45.
3	31.	25.
4	27.	18.
5	16.	18.
6	59.	73.
7	201.	570.
8	380.	1871.
9	558.	1457.
10	452.	611.
11	568.	656.
12	644.	680.
13	688.	725.
14	695.	668.
15	721.	754.
16	897.	733.
17	1287.	804.
18	1466.	702.
19	1110.	736.
20	802.	578.
21	555.	444.
22	419.	323.
23	310.	312.
24	248.	192.
TOTAL	12355.	13083.

5. TRUCK COUNT (LENGTH > 6.5 M) :

HOUR	WESTBOUND (2 LANES)	EASTBOUND (2 LANES)
1	8.	5.
2	6.	2.
3	3.	1.
4	2.	0.
5	0.	0.
6	10.	4.
7	16.	34.
8	48.	167.
9	63.	125.
10	48.	43.
11	57.	32.
12	49.	43.
13	50.	39.
14	53.	49.
15	38.	42.
16	56.	57.
17	77.	42.
18	72.	26.
19	41.	27.
20	17.	15.
21	14.	10.
22	10.	8.
23	8.	7.
24	5.	4.
TOTAL	751.	782.

6. SPEED COUNT :

OVER 130 KM/H	19.
120-130	4.
110-120	6.
100-110	10.
90-100	35.
80- 90	32.
70- 80	169.
60- 70	902.
50- 60	2919.
40- 50	15395.
30- 40	2603.
UNDER 30 KM/H	2194.
TOTAL	24288.

Figure 6.3/ Typical Volume Count Output

6.16.3/ Parameter Index

Routine: 'PCS7'

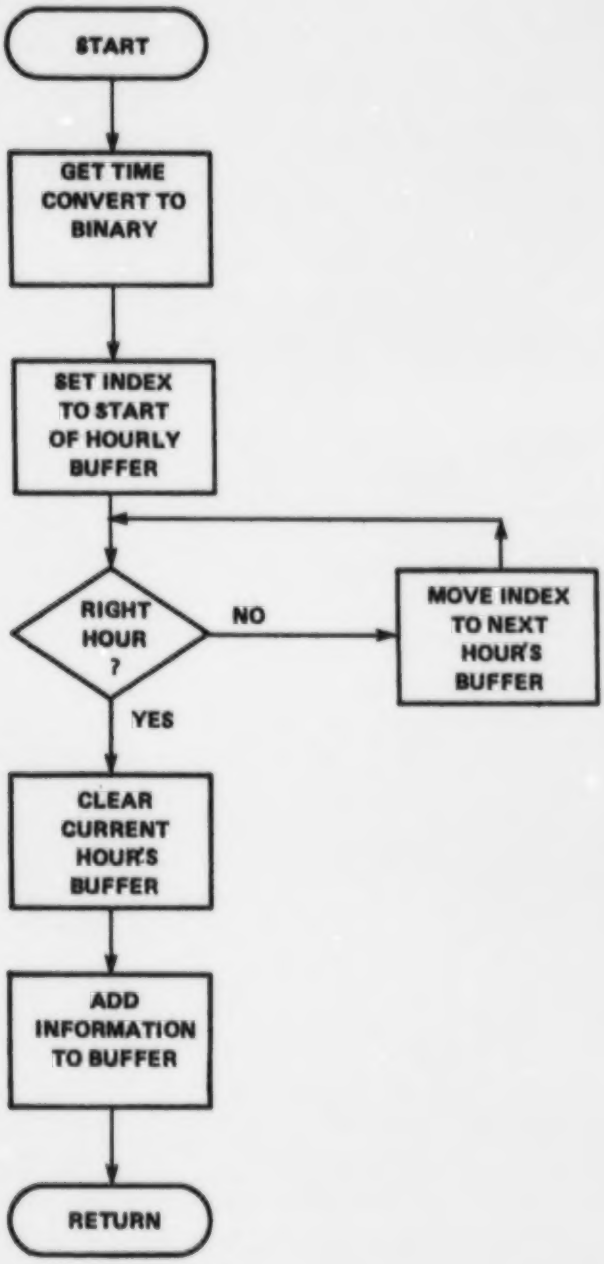
Description: Hourly data transfer routine.

Parameter	Description
HR	- hour of the day in BCD (24-h clock).
G70	- label for internal branch statement.
G71	- label for internal branch statement.
G1	- label for internal branch statement.
G5	- label for internal branch statement.
PTR	- pointer to beginning of volume data.
TEMPØ4	- a temporary variable used for varied applications.
G9	- label for internal branch statement.
SIZE	- amount of memory space required to store all volume counts in both directions for 1 h.

Parameter	Description
TEMPØ5	- a temporary variable used for varied applications.
G51	- label for internal branch statement.
NVOL1	- number of volume counts to be calculated in Direction 1.
TEMPØ1	- a temporary storage area used in varied applications.
TEMPØ3	- a temporary storage area used in varied applications.
TEMPØ6	- a temporary storage area used in varied applications.
TEMPØ7	- a temporary storage area used in varied applications.
VOLDW1	- count descriptor byte for Volume 1.
SPC1	- MSB of index for referencing lane volume data.
SPC2	- LSB of index for referencing lane volume data.
G15	- label for internal branch statement.
G10	- label for internal branch statement.

Parameter	Description
NVOL2	- number of volume counts to be calculated in Direction 2.
G60	- label for internal branch statement.
G45	- label for internal branch statement.
G30	- label for internal branch statement.
G40	- label for internal branch statement.
VD1M	- MSB of hourly volume count.
VD1L	- LSB of hourly volume count.
G52	- label for internal branch statement.
G36	- label for internal branch statement.
G50	- label for internal branch statement.

6.16.4/ Flow Chart



6.17/ 'PCS8'

6.17.1/ Program Listing Reference

The relevant program listing may be found in Appendix A, starting on Page 191, Line 2412, extending to Line 2703, which includes both directions.

6.17.2/ Program Explanation

This routine calculates and classifies vehicular speed and assesses their length. It, like most of the software, is split into two routines, one for each direction. Since these two routines, 'PCS8' and 'PCS8A,' require much mathematical manipulation, they are executed alternately to avoid occupying too much of the alloted cycle time.

The first function of the routine is to assign a limit to the variable 'COUNT6,' which will be used in the exclusion of certain speed data later on in the program. This is followed by a set of instructions which determine if the routine is to be executed this time around. Given that it is, the Direction 1 vehicle 'end' transition timing flag is checked for the first lane. If the flag is set (see 'PCS3/PCS3B'), then the routine knows the vehicle has not yet finished the test section and goes on to the next lane. On the other hand, on finding that the vehicle has completed the section, eligibility is then checked. Since the transition timer is an 8-bit counter with 1/60 of a second increments every 256/60 s, the counter will reset and set the appropriate transition counter roll over flag found in 'C1E.' Therefore, for a lane to be eligible, the roll over flag must be clear to indicate relevant timing data. If the check reveals that the flag is set for the particular channel, then all the appropriate speed and length parameters are cleared. Assuming all is well, the processing continues.

The transition times are loaded and, here again, eligibility is checked. This time, timing values which would result in speeds in excess of 136 mph are eliminated outright. Such values result from unusual vehicle behaviour in and about the test section and tend to clutter up the upper speed classification. Data which are still found to be correct are then manipulated, with the math subroutines, in accordance with the relationship:

$$\text{Speed} = \frac{1}{2}\left[\frac{LS}{BIE} + \frac{LS}{BIS}\right]$$

Similarly, vehicle length is calculated according to :

Length = OCC * speed - LD

Examples of both calculations may be found in Section 2.3. All that is left to be done now is to classify the speed.

The mechanics of grouping the speed are obvious. Each group has a threshold value; therefore, a given speed is compared to these values systematically until the appropriate group is found. As can be seen, in hard copy, both the most and least significant bytes of the threshold values are similarly compared.

The process is repeated for all necessary lanes before control is returned to the calling routine. A printout of a typical speed grouping length count obtained by the telemetry link can be seen in Figure 6.4.

1. PCS STATION I.D. : CONTROLLER #21, STATION #52 -- LAKESHORE
2. HIGHWAY : 2 (EAST OF CREDIT RIVER BRIDGE)
3. DATE : 82/ 9/14, TUESDAY

4. VOLUME COUNT :

HOUR	WESTBOUND (2 LANES)	EASTBOUND (2 LANES)
1	135.	88.
2	86.	45.
3	31.	25.
4	27.	18.
5	16.	18.
6	59.	73.
7	201.	570.
8	380.	1871.
9	558.	1457.
10	452.	611.
11	568.	656.
12	644.	680.
13	688.	725.
14	695.	668.
15	721.	754.
16	897.	733.
17	1287.	804.
18	1466.	702.
19	1110.	736.
20	802.	578.
21	555.	444.
22	419.	323.
23	310.	312.
24	248.	192.
TOTAL	12355.	13083.

5. TRUCK COUNT (LENGTH > 6.5 M) :

HOUR	WESTBOUND (2 LANES)	EASTBOUND (2 LANES)
1	8.	5.
2	6.	2.
3	3.	1.
4	2.	0.
5	0.	0.
6	10.	4.
7	16.	34.
8	48.	167.
9	63.	125.
10	48.	43.
11	57.	32.
12	49.	43.
13	50.	39.
14	53.	49.
15	38.	42.
16	56.	57.
17	77.	42.
18	72.	26.
19	41.	27.
20	17.	15.
21	14.	10.
22	10.	8.
23	3.	7.
24	5.	4.
TOTAL	751.	782.

6. SPEED COUNT :

OVER 130 KM/H	19.
120-130	4.
110-120	6.
100-110	10.
90-100	35.
80- 90	32.
70- 80	169.
60- 70	902.
50- 60	2919.
40- 50	15395.
30- 40	2603.
UNDER 30 KM/H	2194.
TOTAL	24288.

Figure 6.4/ Typical Speed Classification (Speed Limit at 50 km/h)

6.17.3/ Parameter Index

Routine: 'PCS8'

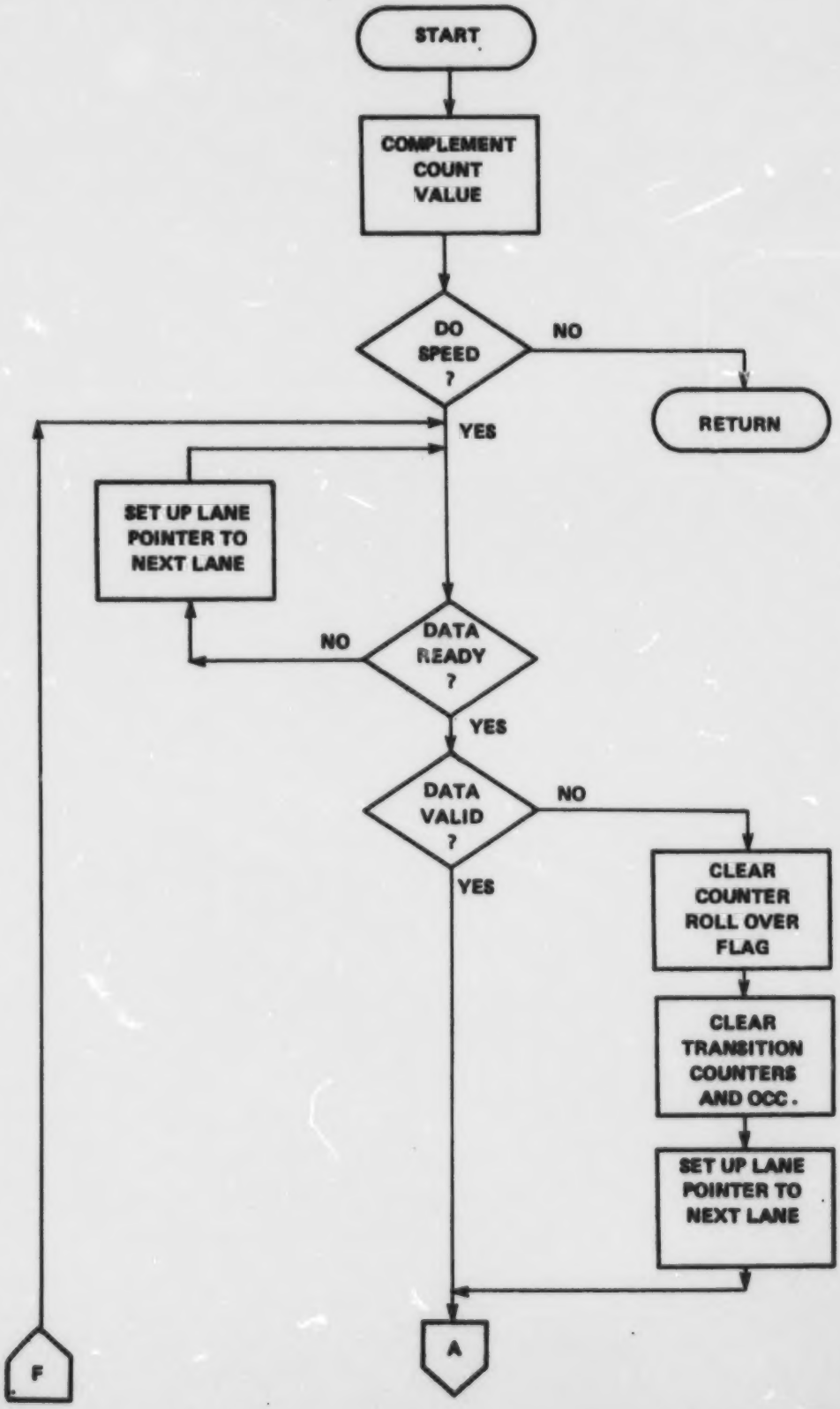
Description: Vehicle speed and length classification.

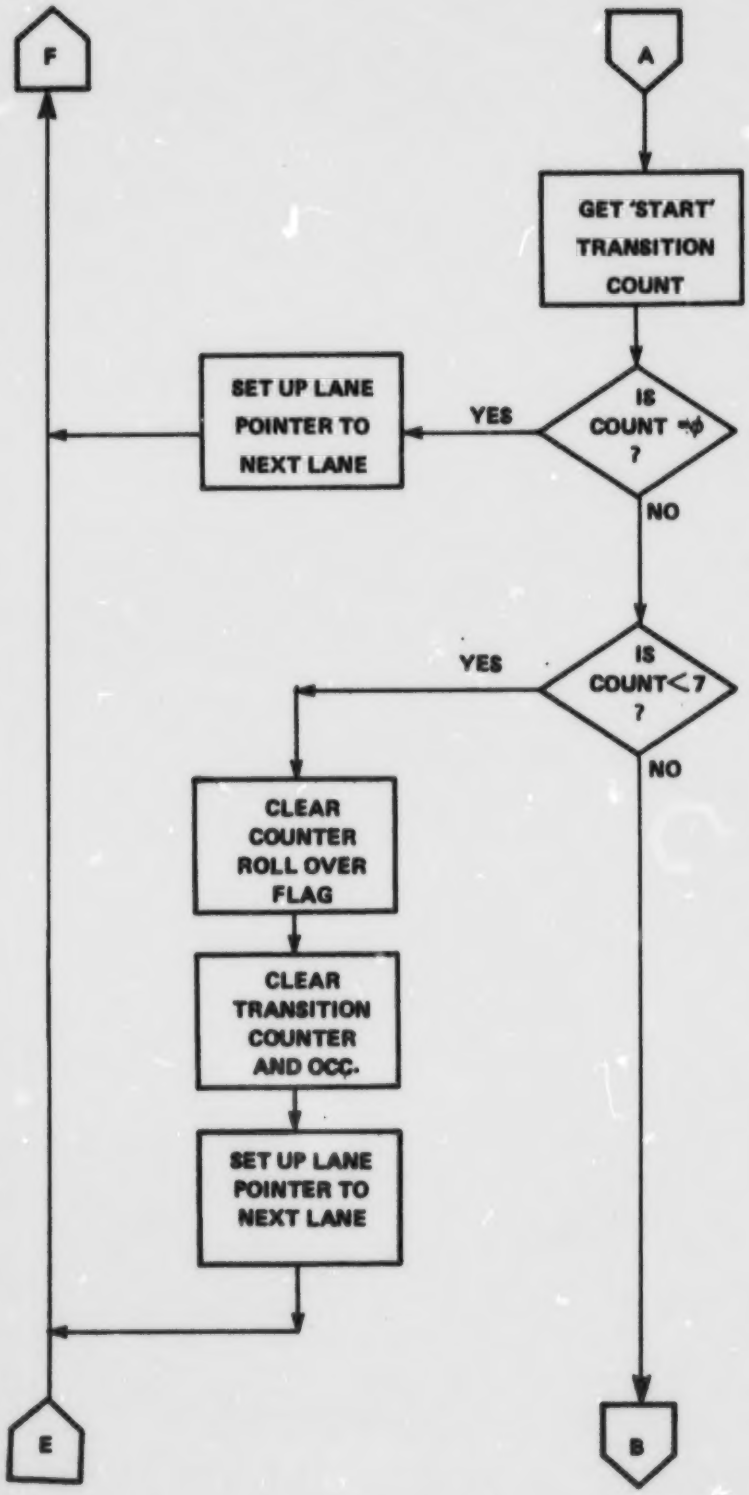
Parameter	Description
COUNT6	- stores the limit of transition time to be a data disqualifier.
COUNT3	- toggling value for alternating speed calculations.
SP99	- label for internal branch statement.
INDEX	- a variable used for the storage of index register values.
COUNT1	- storage area for lane bit mask.
GO	- Direction 1 vehicle end transition timing flag.
SP11	- label for internal branch statement.
C1E	- transition counter roll over flag.
SP49	- label for internal branch statement.
B1E	- 8-bit indexed transition timer for vehicle ends.

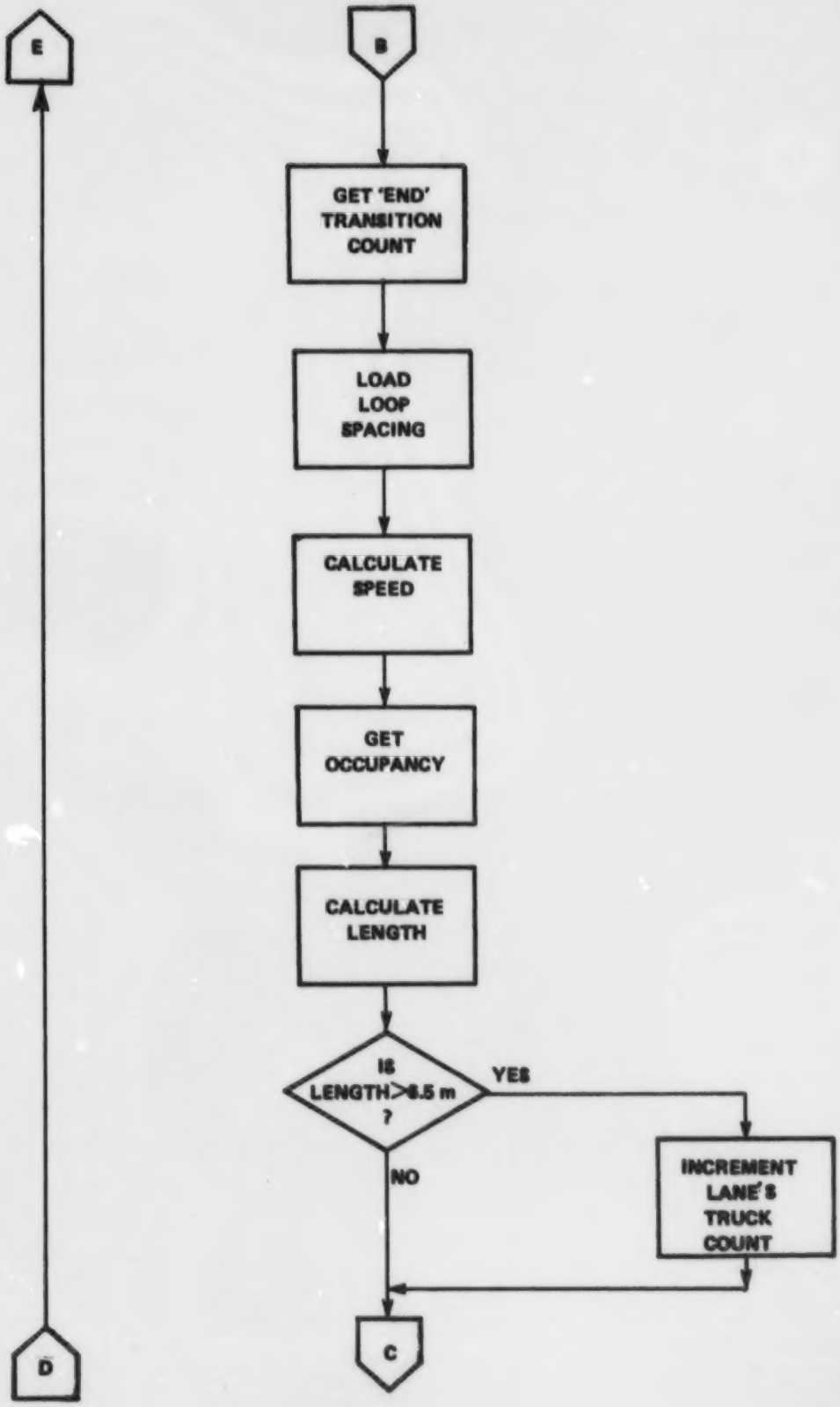
Parameter	Description
OCC	- vehicle occupancy used in length calculation derived by upstream detectors.
B1S	- 8-bit indexed transition timer for vehicle starts.
TEMPØ2	- a temporary storage area used for varied applications.
TEMPØ4	- a temporary storage area used for varied applications.
LS	- variable for loop spacing.
TEMPØ5	- a temporary storage area used for varied applications.
DIV	- subroutine call to invoke the 8-bit division routine.
MULT	- subroutine call to invoke the 8-bit multiplication routine.
LD	- a variable for the loop dimensions.
TRUCK	- the length limit equivalent to 6.5 m.
LENL	- LSB for the storage of the hourly long vehicle count.
LENM	- MSB for the storage of the hourly long vehicle count.

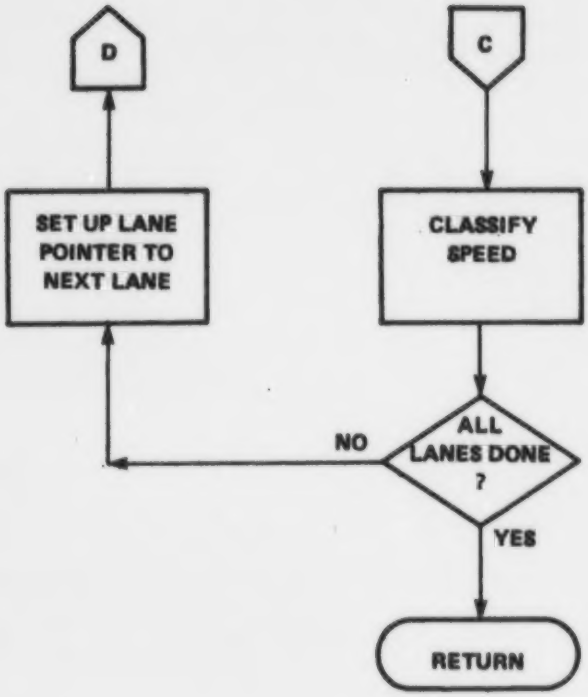
Parameter	Description
SPED	- an indexed value which holds the 12 groups of daily speed totals.
SPD	- an indexed value which represents the MSB of the speed classification thresholds.
REM	- an indexed value which represents the LSB of the speed classification threshold.
NEXTLN	- subroutine call to advance lane data pointer.

6.17.4/ Flow Chart




F
A
GET 'START' TRANSITION COUNT
SET UP LANE POINTER TO NEXT LANE
YES
IS COUNT = ϕ ?
NO
YES
IS COUNT < 7 ?
NO
CLEAR COUNTER ROLL OVER FLAG
CLEAR TRANSITION COUNTER AND OCC.
SET UP LANE POINTER TO NEXT LANE
E
B


E
B
GET 'END' TRANSITION COUNT
LOAD LOOP SPACING
CALCULATE SPEED
GET OCCUPANCY
CALCULATE LENGTH
IS LENGTH>6.5 m ?
YES
NO
INCREMENT LANE'S TRUCK COUNT
C
D


D
C
SET UP LANE POINTER TO NEXT LANE
CLASSIFY SPEED
NO
ALL LANES DONE ?
YES
RETURN

7/ SYSTEM DEVELOPMENT

To every system there are improvements, and to all software there are modifications. PCS is no exception to this; there is literally no end to changes that could be made. Of the myriad of possibilities, some are more beneficial than others.

An improvement that would be very beneficial is an addition to the initialization routine, which would assess the memory size and calculate the maximum number of days to be stored. Along with this, the memory zeroing process could be extended to cover all but the stack, independent of memory size.

An oversight propagated from Phase I software deals with lanes equipped with only one lane detector. This requires the single detector, by definition, to be classified as a downstream detector for software purposes, while wired as an upstream detector for value counts. It is desirable to have such a single detector classified as upstream within the initialization routine, not downstream.

Another desirable feature would be to have a total of all speed data classified and excluded from calculation to compare to volume data.

There are also more detector checking routines that could be employed. The routine used now has provisions for six more failure descriptions. And, of course, there are running and average running total control data not being initialized at all.

A word of caution is in order with respect to modification. The cross-reference table in Appendix A is useful to trace the origin and use of variables. When even minor changes are made to a routine, the implication of it must be traced through the total system. Also, ad hoc modification has its limits. The author is of the opinion that, if a major addition or change proves necessary, the entire software package should be rewritten with those PCS objectives in mind. PCS has ample fourth generation modified software already and, as a result, is becoming less receptive to change.

8/ ACKNOWLEDGEMENTS

A significant contribution in both technical and philosophical matters by Mr. R. Bur is gratefully acknowledged. His continued co-operation in all stages of the project was a key factor in its success.

9/ BIBLIOGRAPHY

[1] Leventhal, Lance A., "6800 Assembly Language Programming," Osborne & Associates, Inc., Berkeley, California, 1978.

[2] Microsystems Group, "M6800 Exorciser User's Guide," Motorola Inc., Phoenix, Arizona, 1975.

[3] Microsystems Group, "Macro Assembler Reference Manual," Motorola Inc., Phoenix, Arizona, 1978.

[4] Microsystems Group, "Co-resident Editor Reference Manual," Motorola Inc., Phoenix, Arizona, 1977.

[5] Microsystems Group, "Exordisk 11/111 Operating System," Motorola Inc., Phoenix, Arizona, 1978.

PAGE

Appendix A/ Equate File 161
Program Listing 162
Cross-Reference Table 193

```
00001          0000  A PCSCAD EQU    $00
00002          0003  A YEAR   EQU    $03
00003          0004  A MONTH  EQU    $04
00004          0005  A DAOFMO EQU    $05
00005          0009  A DAOFWK EQU    $09
00006          000A  A HR     EQU    $0A
00007                  $
00008          0010  A HLANE1 EQU    $10
00009          0011  A HLANE2 EQU    $11
00010          0012  A LDESW1 EQU    $12
00011          0013  A LDESW2 EQU    $13
00012          0014  A HVOL1  EQU    $14
00013          0015  A HVOL2  EQU    $15
00014          0016  A VOLDW1 EQU    $16
00015          001E  A LBNW1  EQU    $1E
00016          001F  A LBNW2  EQU    $1F
00017                  $
00018          006E  A NRAM   EQU    $6E
00019                  $
00020          0088  A DETER1 EQU    $88
00021          0089  A DETER2 EQU    $89
00022          008A  A DETER3 EQU    $8A
00023          008B  A DETER4 EQU    $8B
00024                  $
00025          0070  A VOLDAY EQU    $70
00026          0071  A WW     EQU    $71
00027          0091  A VOBAY1 EQU    $91
00028          0073  A VOBUF1 EQU    $73
00029          0075  A SPDBAY EQU    $75
00030          0076  A SPDAY1 EQU    $76
00031          0078  A SPBUF  EQU    $78
00032          007A  A LENBAY EQU    $7A
00033          007B  A LNBAY1 EQU    $7B
00034          007D  A LNBUF1 EQU    $7D
00035          007F  A DAYMAX EQU    $7F
00036          0095  A SIZE   EQU    $95
00037          0096  A PTRL   EQU    $96
00038          0098  A PTRS   EQU    $98
00039          009B  A BC10CH EQU    $9B
00040          009C  A BC10CL EQU    $9C
00041          009D  A PTR    EQU    $9D
00042                  $
00043          009F  A CYCLE  EQU    $9F
00044                  $
00045          00A0  A SPC1   EQU    $A0
00046          00A1  A SPC2   EQU    $A1
00047          00A2  A KAY2   EQU    $A2
00048          00A3  A TRUCK  EQU    $A3
00049          00A4  A LENGTH EQU    $A4
00050          00A6  A SPEED  EQU    $A6
00051          00A8  A VOBAYL EQU    $A8
00052          00AA  A VOBAYS EQU    $AA
00053          00AC  A DATAON EQU    $AC
00054          00AD  A DATAIN EQU    $AD
00055          00AE  A DETINN EQU    $AE
00056          00AF  A STCHGN EQU    $AF
00057                  $
00058          00B0  A GO     EQU    $B0
```



```
00059          00B1  A GOT    EQU    $B1
00060          00B2  A GOA    EQU    $B2
00061          00B3  A GOTA   EQU    $B3
00062                  $
00063          00BC  A DATAOL EQU    $BC
00064          00BD  A DATAIL EQU    $BD
00065          00BE  A DETINL EQU    $BE
00066          00BF  A STCHOL EQU    $BF
00067          00C0  A DTCTRM EQU    $C0
00068                  $
00069          00E3  A COUNT1 EQU    $E3
00070          00E4  A COUNT2 EQU    $E4
00071          00E5  A COUNT3 EQU    $E5
00072          00E6  A COUNT4 EQU    $E6
00073          00E7  A COUNT5 EQU    $E7
00074          00E8  A COUNT6 EQU    $E8
00075          00EA  A INDEX  EQU    $EA
00076          00EC  A AR1MOM EQU    $EC
00077          00ED  A AR1MOL EQU    $ED
00078          00EE  A AR1MVM EQU    $EE
00079          00EF  A AR1MVL EQU    $EF
00080          00D0  A DTCTRL EQU    $D0
00081                  $
00082          00F2  A XLANE  EQU    $F2
00083          00F3  A PCW1   EQU    $F3
00084          00F4  A KAY1   EQU    $F4
00085          00F5  A TEMP01 EQU    $F5
00086          00F6  A TEMP02 EQU    $F6
00087          00F7  A TEMP03 EQU    $F7
00088          00F8  A TEMP04 EQU    $F8
00089          00F9  A TEMP05 EQU    $F9
00090          00FA  A TEMP06 EQU    $FA
00091          00FB  A TEMP07 EQU    $FB
00092                  $
00093          0000  A OBIX1  EQU    $00
00094          0001  A VSIX1  EQU    $1
00095          0002  A OCMN1M EQU    $2
00096          0003  A OCMN1L EQU    $3
00097          0004  A VMM1   EQU    $4
00098          0005  A OCC1M  EQU    $5
00099          0006  A OCC1L  EQU    $6
00100          0007  A VOL1   EQU    $7
00101          0008  A VB1M   EQU    $8
00102          0009  A VB1L   EQU    $9
00103          000A  A DET1   EQU    $A
00104          000B  A STCHB1 EQU    $B
00105          000C  A LNBIB1 EQU    $C
00106          000D  A LENM   EQU    $D
00107          000E  A LENL   EQU    $E
00108          000F  A OCC    EQU    $F
00109                  $
00110          0000  A B1E    EQU    $0
00111          0001  A B1S    EQU    $1
00112          0002  A B2E    EQU    $2
00113          0003  A B2S    EQU    $3
00114          0004  A LS     EQU    $4
00115          0005  A LSA    EQU    $5
00116          0006  A LD     EQU    $6
```

```
PAGE  003  DEFINE  .BAIO  SUB

00117          0007  A LDA    EQU    $7
00118          0008  A C1E    EQU    $8
00119          0009  A C1B    EQU    $9
00120          000A  A C2E    EQU    $A
00121          000B  A C2B    EQU    $B
00122          0080  A SPD    EQU    $80
00123          0090  A REM    EQU    $90
00124          00A0  A SPED   EQU    $A0
00125                  *
00126          0400  A START  EQU    $400
00127                  *
00128                  * LANE PROPORTIONING CONSTANTS
00129                  *
00130          0000  A PCLN1  EQU    0
00131          0080  A PCLN2  EQU    128
00132          00AB  A PCLN3  EQU    171
00133          00C0  A PCLN4  EQU    192
00134          00CD  A PCLN5  EQU    205
00135          00D5  A PCLN6  EQU    213
00136          00DB  A PCLN7  EQU    219
00137          00E0  A PCLN8  EQU    224
00138                  *
00139                  * FLAG BITS
00140                  *
00141          0001  A TBIT0  EQU    $01
00142          0002  A TBIT1  EQU    $02
00143          0004  A TBIT2  EQU    $04
00144          0008  A TBIT3  EQU    $08
00145          0010  A TBIT4  EQU    $10
00146          0020  A TBIT5  EQU    $20
00147          0040  A TBIT6  EQU    $40
00148          0080  A TBIT7  EQU    $80
00149                  *
00150                  * SUBROUTINE ADDRESSES
00151                  *
00152          C000  A CMP16  EQU    $C000
00153          C030  A NEXTLN EQU    $C030
00154          C060  A SETUP  EQU    $C060
00155          C080  A PRPRTB EQU    $C080
00156          C0B0  A PRPRT  EQU    $C0B0
00157          D000  A INIT   EQU    $D000
00158          D250  A HEADER EQU    $D250
00159          C2A0  A INPUT  EQU    $C2A0
00160          C350  A PCS2   EQU    $C350
00161          D4A0  A PCS3   EQU    $D4A0
00162          C400  A PCS4   EQU    $C400
00163          C4A0  A PCS5   EQU    $C4A0
00164          C550  A PCS6   EQU    $C550
00165          C6A0  A PCS7   EQU    $C6A0
00166          C840  A PCS4A  EQU    $C840
00167          C8D0  A PCS5A  EQU    $C8D0
00168          C950  A PCS6A  EQU    $C950
00169          C9B0  A MULT   EQU    $C9B0
00170          C9E0  A DIV    EQU    $C9E0
00171          CA00  A INPUTT EQU    $CA00
00172          D550  A PCS3A  EQU    $D550
00173          CDB0  A PCS3B  EQU    $CDB0
00174          CC00  A PCS3C  EQU    $CC00
```

```
PAGE  004  DEFINE  .BAIO  SUB

00175          CC50  A PCS8   EQU    $CC50
00176          CDA0  A PCS8A  EQU    $CDA0
00177          CF00  A PCS7A  EQU    $CF00
00178                  *
00179                  * INPUT PORTS
00180                  *
00181          5001  A PORT1  EQU    $5001    DETECTORS 01-08 FOR DIRECTION ONE
00182          5002  A PORT2  EQU    $5002    DETECTORS 09-016 FOR DIRECTION ONE
00183          5003  A PORT3  EQU    $5003    DETECTORS 01-08 FOR DIRECTION TWO
00184          5004  A PORT4  EQU    $5004    DETECTORS 09-016 FOR DIRECTION TWO
00185
00186                  *
00187                  *         MOTOROLA M68SAM CROSS-ASSEMBLER PAGE 1
00188                  *
00189                  * M68SAM IS THE PROPERTY OF MOTOROLA SPD, INC.
00190                  * CALTRANS TO N.T.C. MODIFIED, RELEASE 1.0
00191                  *
00192                  * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
00193                  *
00194                  *      1981 08 25
00195                  *
00196                  *    RELEASE LEVEL 6.75
00197                  *
00198                         MAN    SUB
00199                  *
00200                  * CMP16. AS REV. 1.50
00201                  * STEVEN SKELTON
00202                  *
00203                  * CMP16 IS A GENERALIZED SUBROUTINE TO COMPARE
00204                  * TWO 16 BIT NUMBERS.
00205                  *
00206                  * CALLING SEQUENCE
00207                  *
00208                  *     1. A 16 BIT NUMBER IS STORED @ TEMP02(MSB) & TEMP03(LSB)
00209                  *     2. THE X REGISTER IS POINTED TOWARDS TWO LOCATIONS
00210                  *        (M=MSB,M+1=LSB)
00211                  *     3. JSR CMP16
00212                  *
00213                  * THE ROUTINE WILL RETURN TO CALLER WITH:
00214                  *    1. CARRY SET IF (M,M+1)>(TEMP02,03)
00215                  *    2. CARRY CLEAR OTHERWISE
00216                  *
00217                  *         MOTOROLA M68SAM CROSS-ASSEMBLER PAGE 2
00218                  *
00219A C000                      ORG    CMP16
00220                  *
00221A C000 36                   PSHA              SAVE A
00222A C001 A6 00    A           LDAA   0,X        GET INDEXED MSB
00223A C003 91 F6    A           CMPA   TEMP02     > MEMORY MSB?
00224A C005 22 0B C012           BHI    ONEBIG     YES -- SET (INDEXED LOC) > (TEMP LOC) FLAG
00225A C007 26 06 C00F           BNE    ONESML     NO, SMALLER -- CLEAR FLAG
00226A C009 A6 01    A           LDAA   1,X        GET INDEXED LSB
00227A C00B 91 F7    A           CMPA   TEMP03     > MEMORY LSB
00228A C00D 22 03 C012           BHI    ONEBIG     YES -- SET FLAG
00229A C00F 0C              ONESML CLC              CLEAR FLAG
00230A C010 32                   PULA
00231A C011 39                   RTS
00232A C012 0D              ONEBIG SEC              SET THE FLAG
```

```
00233A C013 32                  PULA
00234A C014 39                  RTS
00235                   *
00236                   * NEXTLN SUBROUTINE
00237                   *
00238                   * CORRECT INDEXES FOR NEXT MAIN LANE DATA, CHECK FOR LAST LANE
00239                   *
00240A C030                     ORG     NEXTLN
00241A C030 36                  PSHA            SAVE ACC. A
00242A C031 96 E8     A         LDAA    INDEX+1 GET MAIN LANE LSB INDEX -- LAST LANE?
00243A C033 27 09 C03E          BEQ     LSTLN   YES
00244A C035 80 10     A         SUBA    #010    MOVE INDEX TO NEXT LANE
00245A C037 97 E8     A         STAA    INDEX+1 STORE IT
00246A C039 32                  PULA            RESTORE A
00247A C03A DE EA     A         LDX     INDEX   RESET X REG
00248A C03C 0C                  CLC             CLEAR LAST LANE FLAG
00249A C03D 39                  RTS
00250A C03E 32          LSTLN   PULA            RESTORE A
00251A C03F 0D                  SEC             SET LAST LANE FLAG
00252A C040 39                  RTS
00253                   *
00254                   *          MOTOROLA M68SAM CROSS-ASSEMBLER PAGE 3
00255                   *
00256                   *                     SETUP SUBROUTINE
00257                   *
00258                   * SET DATA INDEXES FOR MAIN LANE DATA ACCESS
00259                   *
00260A C040                     ORG     SETUP
00261A C040 CE 0170   A         LDX     #00170  SET INDEX AT HIGHEST LANE - 1
00262A C043 DF EA     A         STX     INDEX   & SAVE IT
00263A C045 39                  RTS
00264                   *
00265                   *          MOTOROLA M68SAM CROSS-ASSEMBLER PAGE 1
00266                   *
00267                   * M68SAM IS THE PROPERTY OF MOTOROLA SPD, INC.
00268                   *        COPYRIGHT 1974 BY MOTOROLA INC
00269                   *
00270                   * MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
00271                   * CALTRANS DISTRICT 11 MODIFIED, RELEASE 1.7
00272                   *
00273                   * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
00274                   *
00275                   *          10/ 4/1979        13: 41: 41
00276                   *
00277                   *     RELEASE LEVEL 6.75
00278                   *
00279                           NAM     PRPRT
00280                   *
00281                   * PRPRT.AS  REV. 0.02
00282                   * 2/27/74
00283                   *
00284                   * THIS SUB-ROUTINE PERFORMS THE OPERATION
00285                   *
00286                   *      N=N(1-PCW/256)
00287                   *
00288                   * WHERE
00289                   *
00290                   *       N=A 8 OR 16 BIT NUMBER POINTED TO BY (X) OR (X), (X+1)
```

```
PAGE  006  TRY     .SA10  PRPRT

                        *       PCW=PROPORTION OF N TO BE SUBTRACTED
                        *
                        * CALLING SEQUENCE
                        *
                        *       PLACE PCW IN ACCA
                        *
                        *          MOTOROLA M68SAM CROSS-ASSMBLER PAGE 2
                        *
                        * SET THE INDEX REGISTER TO POINT TO N OR N - MSB
                        * JSR PRPRTS OR PRPRT
                        *
                        * ON RETURN (ACCA)=PROPORTION OF N REQUIRED - MSB
                        * AND (ACCB)=PROPORTION OF N REQUIRED - LSB
                        *
A C080                          ORG     PRPRTS
                        *
A C080 36                       PSHA                PCW ON STACK
A C081 E6 00      A             LDAB    X           ORIGINAL VALUE
A C083 4F                       CLRA                CLEAR MSB
A C084 20 2F C0B5               BRA     P10         ON WITH PROGRAM
                        *
A C0B0                          ORG     PRPRT
A C0B0 36                       PSHA                PCW ON STACK
A C0B1 A6 00      A             LDAA    X           ORIGINAL VALUE - MSB
A C0B3 E6 01      A             LDAB    1,X         ORIGINAL VALUE - LSB
A C0B5 36               P10     PSHA                PUT ON STACK
A C0B6 37                       PSHB                PUT ON STACK
A C0B7 34                       DES                 SPACE FOR ROUND-OFF FLAG
A C0B8 30                       TSX                 POINT INDEX REG. TO STACKED DATA
                        *
                        * THE STACK NOW LOOKS LIKE THIS!
                        *
                        *  +0  ROUND OFF FLAG
                        *  +1  ORIGINAL VALUE - LSB
                        *  +2  ORIGINAL VALUE - MSB
                        *  +3  PROPORTION CONTROL WORD
                        *
A C0B9 6F 00      A AGAIN  CLR     0,X         CLEAR ROUND OFF FLAG
                        *
                        *          MOTOROLA M68SAM CROSS-ASSMBLER PAGE 3
                        *
A C0BB 64 02      A             LSR     2,X         SHIFT MSB
A C0BD 66 01      A             ROR     1,X         SHIFT LSB
A C0BF 24 02 C0C3               BCC     P1          IF NO CARRY, GO ON
A C0C1 6C 00      A             INC     0,X         SET ROUND-OFF FLAG
                        *
A C0C3 68 03      A P1          ASL     3,X         CHECK FOR PCW BIT SET
A C0C5 24 0D C0D4               BCC     CHECK       IF NO BIT SET, DON'T SUBTRACT
A C0C7 E0 01      A             SUBB    1,X         SUBTRACT PROPORTION - LSB
A C0C9 A2 02      A             SBCA    2,X         SUBTRACT PROPORTION - MSB
A C0CB 6D 00      A             TST     0,X         IS ROUND-OFF FLAG SET?
A C0CD 27 05 C0D4               BEQ     CHECK       IF NOT, CHECK TO SEE IF RETURN
A C0CF C0 01      A             SUBB    #01         SUBTRACT ONE FROM LSB
A C0D1 24 01 C0D4               BCC     CHECK       IF NO OVERFLOW, CHECK ON RETURN
A C0D3 4A                       DECA                SUBTRACT OVERFLOW FROM MSB
                        *
A C0D4 6D 03      A CHECK  TST     3,X         CHECK PCW FOR ANY BIT SET!
A C0D6 26 E1 C0B9               BNE     AGAIN       IF ONE MORE BIT SET, GO BACK THROUGH
```

```
PAGE  007  TRY     .SAIO  PRPRT

00349A C0B8 31                INS
00350A C0B9 31                INS
00351A C0BA 31                INS
00352A C0BB 31                INS
00353A C0BC 39                RTS
00354                   *
00355                   *         MOTOROLA M6800 CROSS-ASSEMBLER PAGE 1
00356                   *
00357                   * M6800 IS THE PROPERTY OF MOTOROLA SPD, INC.
00358                   * COPYRIGHT 1974 BY MOTOROLA INC
00359                   *
00360                   * MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
00361                   * WRITTEN BY M.T.C. , RELEASE 2.0
00362                   *
00363                   * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
00364                   *
00365                   *       1982 03 17
00366                   *
00367                   *    RELEASE LEVEL 6.75
00368                   *
00369                         NAM    MATH
00370                   *
00371                   * MATH AS  REV. 0.00
00372                   * STEVEN SKELTON
00373                   *
00374                   * THIS ROUTINE PROVIDES SUBROUTINES FOR MULTIPLICATION AND DIVISION.
00375                   *
00376                   * MULTIPLIER AND MULTIPLICAND ARE ASSIGNED TO TEMP04,TEMP05.
00377                   * THE PRODUCT IS ASSIGNED TO:
00378                   *                                  (A) MSB ==> TEMP06
00379                   *                                  (B) LSB ==> TEMP07
00380                   *
00381A C980                   ORG    MULT
00382                   *
00383A C980 4F                CLRA
00384A C981 5F                CLRB
00385A C982 CE 0008  A        LDX    #8
00386A C985 58          MM1   ASLB
00387A C986 49                ROLA
00388A C987 78 00F8  A        ASL    TEMP04
00389A C98A 24 04 C9C0        BCC    MM2
00390A C98C DB F9    A        ADDB   TEMP05
00391A C98E 89 00    A        ADCA   #0
00392A C9C0 09          MM2   DEX
00393A C9C1 26 F2 C985        BNE    MM1
00394A C9C3 97 FA    A        STAA   TEMP06
00395A C9C5 D7 FB    A        STAB   TEMP07
00396A C9C7 39                RTS
00397                   *
00398                   *
00399                   * THE DIVISOR AND DIVIDEND ARE ASSIGNED TO TEMP04,TEMP05 RESPECTIVLY.
00400                   * WHILE THE QUOTIENT IS TEMP06 AND ANY REMAINDER IS IN TEMP07.
00401                   *
00402                   *
00403A C9E0                   ORG    DIV
00404                   *
00405A C9E0 CE 0008  A        LDX    #8
00406A C9E3 4F                CLRA
```

```
PAGE  008  TRY     .BAIO  MATH

00407A C9E4 B6 F9     A         LDAB   TEMP05
00408A C9E6 58          MD1     ASLB
00409A C9E7 49                  ROLA
00410A C9E8 91 F8     A         CMPA   TEMP04
00411A C9EA 25 03 C9EF          BCS    MD2
00412A C9EC 90 F8     A         SUBA   TEMP04
00413A C9EE 5C                  INCB
00414A C9EF 09          MD2     DEX
00415A C9F0 26 F4 C9E6          BNE    MD1
00416A C9F2 B7 FA     A         STAB   TEMP06
00417A C9F4 97 FB     A         STAA   TEMP07
00418A C9F6 39                  RTS
00419                   *
00420                   *         MOTOROLA MASBAR CROSS-ASSEMBLER PAGE 1
00421                   *
00422                   * MASBAR IS THE PROPERTY OF MOTOROLA SPD, INC.
00423                   * WRITTEN BY M.T.C. , RELEASE 2.0
00424                   *
00425                   * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
00426                   *
00427                   *       1981 08 26
00428                   *
00429                   *    RELEASE LEVEL 4.75
00430                   *
00431                           MAM    START
00432                   *
00433                   * START. AS REV. 1.00
00434                   * STEVEN SKELTON
00435                   *
00436A D000                     ORG    INIT
00437                   *
00438                   *"START" IS AN INITIALISING ROUTINE WITH THREE FUNCTIONS
00439                   *
00440                   *       (1) TO ZERO MEMORY
00441                   *       (2) TO DEFINE AND CALCULATE VARIABLES
00442                   *       (3) AND TO OUTPUT THE FIRST HEADER
00443                   *
00444A D000 CE 0080   A         LDX    0080
00445A D003 4F                  CLRA
00446A D004 A7 00     A D1      STAA   0,X
00447A D006 08                  INX
00448                   *
00449                   *SMALL MEMORY ZEROING
00450                   *
00451A D007 BF F3     A         STX    TEMP01
00452A D009 B6 F3     A         LDAB   TEMP01
00453A D00B C0 07     A         SUBB   0007
00454A D00D 26 F5 D004          BNE    D1
00455A D00F B6 F6     A         LDAB   TEMP02
00456A D011 C0 BF     A         SUBB   00BF
00457A D013 26 EF D004          BNE    D1
00458                   *
00459A D015 CE 0100   A         LDX    00100
00460A D018 86 01     A         LDAA   0001
00461A D01A 95 1E     A C10     BITA   LDNW1
00462A D01C 26 02 D020          BNE    C13
00463A D01E A7 0C     A         STAA   LNDIS1,X
00464A D020 BF F5     A C13     STX    TEMP01
```

```
PAGE  009  TRY     .BAIO  START

00465A D022 C6 10     A         LDAB   0010
00466A D024 BB F6     A         ADDB   TEMP02
00467A D026 B7 F6     A         STAB   TEMP02
00468A D028 BE F5     A         LDX    TEMP01
00469A D02A 48                  ASLA
00470A D02B 24 EB D01A          BCC    C10
00471                   *
00472A D02D CE 0100   A         LDX    00100
00473A D030 86 01     A         LDAA   0001
00474A D032 95 1F     A C12     BITA   LDNW2
00475A D034 26 02 D038          BNE    C11
00476A D036 A7 0C     A         STAA   LNDIS1+080,X
00477A D038 BF F5     A C11     STX    TEMP01
00478A D03A C6 10     A         LDAB   0010
00479A D03C BB F6     A         ADDB   TEMP02
00480A D03E B7 F6     A         STAB   TEMP02
00481A D040 BE F5     A         LDX    TEMP01
00482A D042 48                  ASLA
00483A D043 24 EB D032          BCC    C12
00484                   *
00485                   * VARIABLE DEFINITION
00486                   *
00487A D045 96 1E     A RET     LDAA   LDNW1
00488A D047 97 C0     A         STAA   DTCTRM
00489A D049 96 12     A         LDAA   LDESW1
00490A D04B 97 E0     A         STAA   DTCTRM+020
00491A D04D 96 1F     A         LDAA   LDNW2
00492A D04F 97 D0     A         STAA   DTCTRL
00493A D051 96 13     A         LDAA   LDESW2
00494A D053 97 F0     A         STAA   DTCTRL+020
00495A D055 86 01     A         LDAA   0001
00496A D057 97 70     A         STAA   VOLDAY
00497A D059 97 E6     A         STAA   COUNT4
00498A D05B 97 75     A         STAA   SPDDAY
00499A D05D 97 7A     A         STAA   LENDAY
00500A D05F 97 9F     A         STAA   CYCLE
00501A D061 97 F4     A         STAA   KAY1
00502A D063 97 A2     A         STAA   KAY2
00503A D065 86 1A     A         LDAA   001A
00504A D067 97 F3     A         STAA   PCW1
00505A D069 7F 0073   A         CLR    VOBUF1
00506A D06C CE 0300   A         LDX    00300
00507A D06F BF 76     A         STX    SPDAY1
00508A D071 CE 0028   A         LDX    00028
00509A D074 BF 78     A         STX    SPBUF
00510A D076 86 07     A         LDAA   0007
00511A D078 97 98     A         STAA   DC10CN
00512A D07A CE 0300   A         LDX    00300
00513A D07D BF A6     A         STX    SPEED
00514                   *
00515A D07F 86 AC     A         LDAA   00AC
00516A D081 97 A3     A         STAA   TRUCK
00517A D083 86 05     A         LDAA   0005
00518A D085 97 7F     A         STAA   DAYMAX
00519                   *
00520A D087 CE 0200   A         LDX    00200
00521A D08A 86 0F     A         LDAA   000F
00522A D08C C6 30     A         LDAB   0030
```

```
PAGE  010  TRY     .BAIO  START

00523A D08E A7 80    A        STAA   SPD,X
00524A D090 E7 90    A        STAB   SPD+010,X
00525                  B
00526A D092 4A                DECA
00527A D093 C6 24    A        LDAB   #024
00528A D095 A7 81    A        STAA   SPD+1,X
00529A D097 E7 91    A        STAB   SPD+011,X
00530                  B
00531A D099 4A                DECA
00532A D09A C6 18    A        LDAB   #018
00533A D09C A7 82    A        STAA   SPD+2,X
00534A D09E E7 92    A        STAB   SPD+012,X
00535                  B
00536A D0A0 4A                DECA
00537A D0A1 C6 0C    A        LDAB   #00C
00538A D0A3 A7 83    A        STAA   SPD+3,X
00539A D0A5 A7 93    A        STAB   SPD+013,X
00540                  B
00541A D0A7 4A                DECA
00542A D0A8 4A                DECA
00543A D0A9 C6 16    A        LDAB   #016
00544A D0AB A7 84    A        STAA   SPD+4,X
00545A D0AD E7 94    A        STAB   SPD+014,X
00546                  B
00547A D0AF 4A                DECA
00548A D0B0 C6 29    A        LDAB   #029
00549A D0B2 A7 85    A        STAA   SPD+5,X
00550A D0B4 E7 95    A        STAB   SPD+015,X
00551                  B
00552A D0B6 4A                DECA
00553A D0B7 C6 1B    A        LDAB   #01B
00554A D0B9 A7 86    A        STAA   SPD+6,X
00555A D0BB E7 96    A        STAB   SPD+016,X
00556                  B
00557A D0BD 4A                DECA
00558A D0BE C6 12    A        LDAB   #012
00559A D0C0 A7 87    A        STAA   SPD+7,X
00560A D0C2 E7 97    A        STAB   SPD+017,X
00561                  B
00562A D0C4 4A                DECA
00563A D0C5 C6 06    A        LDAB   #06
00564A D0C7 A7 88    A        STAA   SPD+8,X
00565A D0C9 E7 98    A        STAB   SPD+018,X
00566                  B
00567A D0CB 4A                DECA
00568A D0CC 4A                DECA
00569A D0CD C6 36    A        LDAB   #036
00570A D0CF A7 89    A        STAA   SPD+9,X
00571A D0D1 E7 99    A        STAB   SPD+019,X
00572                  B
00573A D0D3 4A                DECA
00574A D0D4 C6 24    A        LDAB   #024
00575A D0D6 A7 8A    A        STAA   SPD+0A,X
00576A D0D8 E7 9A    A        STAB   SPD+01A,X
00577                  B
00578A D0DA C6 10    A        LDAB   #010
00579A D0DC CE 0200  A        LDX    #00200
00580A D0DF 86 D0    A        LDAA   #0D0
```

```
PAGE  011  TRY     .BAIO  START

00581A D0E1 A7 04    A C8B    STAA   LB,X
00582A D0E3 A7 05    A        STAA   LBA,X
00583A D0E5 DF F3    A        STX    TEMP01
00584A D0E7 DB F6    A        ADDB   TEMP02
00585A D0E9 D7 F6    A        STAB   TEMP02
00586A D0EB C6 10    A        LDAB   #010
00587A D0ED DE F3    A        LDX    TEMP01
00588A D0EF BC 0200  A        CPX    #00200
00589A D0F2 26 ED D0E1        BNE    C8B
00590                  B
00591A D0F4 CE 0200  A        LDX    #00200
00592A D0F7 86 30    A        LDAA   #030
00593A D0F9 A7 06    A C87    STAA   LD,X
00594A D0FB A7 07    A        STAA   LDA,X
00595A D0FD DF F3    A        STX    TEMP01
00596A D0FF DB F6    A        ADDB   TEMP02
00597A D101 D7 F6    A        STAB   TEMP02
00598A D103 C6 10    A        LDAB   #010
00599A D105 DE F3    A        LDX    TEMP01
00600A D107 BC 0200  A        CPX    #00200
00601A D10A 26 ED D0F9        BNE    C87
00602                  B
00603                  B VARIABLE COMPUTATION
00604                  B
00605A D10C 4F                CLRA
00606A D10D 97 F3    A        STAA   TEMP01
00607A D10F D6 16    A        LDAB   VOLDW1
00608A D111 27 01 D114        BEQ    C2
00609A D113 4C                INCA
00610A D114 D6 17    A C2     LDAB   VOLDW1+1
00611A D116 27 01 D119        BEQ    C3
00612A D118 4C                INCA
00613A D119 D6 18    A C3     LDAB   VOLDW1+2
00614A D11B 27 01 D11E        BEQ    C4
00615A D11D 4C                INCA
00616A D11E D6 19    A C4     LDAB   VOLDW1+3
00617A D120 27 01 D123        BEQ    C5
00618A D122 4C                INCA
00619A D123 D6 1A    A C5     LDAB   VOLDW1+4
00620A D125 27 01 D128        BEQ    C6
00621A D127 4C                INCA
00622A D128 D6 1B    A C6     LDAB   VOLDW1+5
00623A D12A 27 01 D129        BEQ    C7
00624A D12C 4C                INCA
00625A D129 D6 1C    A C7     LDAB   VOLDW1+6
00626A D12F 27 01 D132        BEQ    C18
00627A D131 4C                INCA
00628A D132 D6 1D    A C18    LDAB   VOLDW1+7
00629A D134 27 01 D137        BEQ    C19
00630A D136 4C                INCA
00631A D137 48         C19    ASLA
00632A D138 97 95    A        STAA   SIZE
00633                  B
00634A D13A 48                ASLA
00635A D13B 48                ASLA
00636A D13C 48                ASLA
00637A D13D 97 F3    A        STAA   TEMP01
00638A D13F 9B F3    A        ADDA   TEMP01
```

```
PAGE  012  TRY      .SAIO  START

00639A B141 0C                  CLC
00640A B142 99 F5     A         ADCA   TEMP01
00641A B144 24 04 B14A          BCC    C9
00642A B146 C6 01     A         LDAB   #001
00643A B148 B7 73     A         STAB   VSBUF1
00644A B14A 8B 04     A C9      ADDA   #04
00645A B14C 97 74     A         STAA   VSBUF1+1
00646                   *
00647A B14E B6 73     A         LDAB   VSBUF1
00648A B150 B7 F8     A         STAB   TEMP04
00649A B152 C6 05     A         LDAB   #5
00650A B154 B7 F9     A         STAB   TEMP05
00651A B156 BB C9B0   A         JBR    MULT
00652A B159 B6 FB     A         LDAB   TEMP07
00653A B15B B7 F7     A         STAB   TEMP03
00654A B15D 96 74     A         LDAA   VSBUF1+1
00655A B15F 97 F8     A         STAA   TEMP04
00656A B161 BB C9B0   A         JBR    MULT
00657A B164 96 FB     A         LDAA   TEMP07
00658A B166 97 A8     A         STAA   LENGTH+1
00659A B168 B6 F7     A         LDAB   TEMP03
00660A B16A BB FA     A         ADDB   TEMP06
00661A B16C CB 04     A         ADDB   #04          COMPENSATE FOR #START BEING 0400
00662A B16E B7 A4     A         STAB   LENGTH
00663                   *
00664A B170 BE 73     A         LDX    VSBUF1
00665A B172 BF 7B     A         STX    LNBUF1
00666A B174 CE 0400   A         LDX    #START
00667A B177 BF 91     A         STX    VSDAY1
00668A B179 BF 71     A         STX    WW
00669                   *
00670A B17B 08                  INX
00671A B17C 08                  INX
00672A B17D 08                  INX
00673A B17E 08                  INX
00674A B17F BF 9B     A         STX    PTR
00675                   *
00676A B181 BE A4     A         LDX    LENGTH
00677A B183 BF A8     A         STX    VSDAYL
00678A B185 BF 7B     A         STX    LNDAY1
00679A B187 08                  INX
00680A B188 08                  INX
00681A B189 08                  INX
00682A B18A 08                  INX
00683A B18B BF 96     A         STX    PTRL
00684                   *
00685A B18D BE A6     A         LDX    SPEED
00686A B18F BF AA     A         STX    VSDAYS
00687A B191 08                  INX
00688A B192 08                  INX
00689A B193 08                  INX
00690A B194 08                  INX
00691A B195 BF 98     A         STX    PTRS
00692                   *
00693                   * INITIAL HEADER
00694                   *
00695A B197 BE 91     A         LDX    VSDAY1
00696A B199 96 03     A         LDAA   YEAR
```

```
PAGE  013  TRY      .SAIO  START

00697A B19B A7 00     A         STAA   0,X
00698A B19D 96 04     A         LDAA   MONTH
00699A B19F A7 01     A         STAA   1,X
00700A B1A1 96 05     A         LDAA   DAOFMO
00701A B1A3 A7 02     A         STAA   2,X
00702A B1A5 96 09     A         LDAA   DAOFWK
00703A B1A7 A7 03     A         STAA   3,X
00704                   *
00705A B1A9 BE A8     A         LDX    VSDAYL
00706A B1AB 96 03     A         LDAA   YEAR
00707A B1AD A7 00     A         STAA   0,X
00708A B1AF 96 04     A         LDAA   MONTH
00709A B1B1 A7 01     A         STAA   1,X
00710A B1B3 96 05     A         LDAA   DAOFMO
00711A B1B5 A7 02     A         STAA   2,X
00712A B1B7 96 09     A         LDAA   DAOFWK
00713A B1B9 A7 03     A         STAA   3,X
00714                   *
00715A B1BB BE AA     A         LDX    VSDAYS
00716A B1BD 96 03     A         LDAA   YEAR
00717A B1BF A7 00     A         STAA   0,X
00718A B1C1 96 04     A         LDAA   MONTH
00719A B1C3 A7 01     A         STAA   1,X
00720A B1C5 96 05     A         LDAA   DAOFMO
00721A B1C7 A7 02     A         STAA   2,X
00722A B1C9 96 09     A         LDAA   DAOFWK
00723A B1CB A7 03     A         STAA   3,X
00724                   *
00725A B1CB 39                  RTS
00726                   *
00727                   *         MOTOROLA M68SAM CROSS-ASSMBLER PAGE 1
00728                   *
00729                   * M68SAM IS THE PROPERTY OF MOTOROLA SPD, INC.
00730                   * WRITTEN BY M.T.C. , RELEASE 2.0
00731                   *
00732                   * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
00733                   *
00734                   *      1981 08 26
00735                   *
00736                   *    RELEASE LEVEL 6.75
00737                   *
00738                           NAM    HEADER
00739                   *
00740                   * HEADER. AS REV. 1.00
00741                   *
00742                   *STEVEN SKELTON
00743                   *
00744                   *'HEADER' PROVIDES THE DAY, DATA, AND HEADER INFORMATION
00745                   *
00746A B250                     ORG    HEADER
00747                   *
00748A B250 C6 0C     A         LDAB   #00C
00749A B252 B7 F7     A         STAB   TEMP03
00750A B254 CE 0200   A         LDX    #0200
00751A B257 BF FA     A         STX    TEMP06
00752A B259 A6 A0     A H31     LDAA   SPED,X
00753A B25B 6F A0     A         CLR    SPED,X
00754A B25D BE 98     A         LDX    PTRS
```

```
00755A B2BF A7 00     A        STAA   0,X
00756A B2C1 7C 0099   A        INC    PTR8+1
00757                    E
00758A B2C4 BE FA     A        LDX    TEMP06
00759A B2C6 A6 D0     A        LDAA   SPED+010,X
00760A B2C8 6F D0     A        CLR    SPED+010,X
00761A B2CA BE 98     A        LDX    PTR8
00762A B2CC A7 00     A        STAA   0,X
00763A B2CE 7C 0099   A        INC    PTR8+1
00764                    E
00765A B2D1 BE FA     A        LDX    TEMP06
00766A B2D3 A6 C0     A        LDAA   SPED+020,X
00767A B2D5 6F C0     A        CLR    SPED+020,X
00768A B2D7 BE 98     A        LDX    PTR8
00769A B2D9 A7 00     A        STAA   0,X
00770                    E
00771A B2DB 7A 00F7   A        DEC    TEMP03
00772A B2DE 7D 00F7   A        TST    TEMP03
00773A B2E1 27 0A B2ED         BEQ    H20
00774A B2E3 7C 00FD   A        INC    TEMP07
00775A B2E6 7C 0099   A        INC    PTR8+1
00776A B2E9 BE FA     A        LDX    TEMP06
00777A B2EB 20 CC B2B9         BRA    H21
00778                    E
00779A B2ED 0C           H20   CLC
00780                    E
00781A B2EE 96 9F     A        LDAA   CYCLE
00782A B2F0 4C                 INCA
00783A B2F1 97 9F     A        STAA   CYCLE
00784A B2F3 96 70     A        LDAA   VOLDAY
00785A B2F5 80 05     A        SUBA   #005
00786A B2F7 27 08 B2A1         BEQ    H9
00787A B2F9 96 9F     A        LDAA   CYCLE
00788A B2FB 97 70     A        STAA   VOLDAY
00789A B2FD 97 72     A        STAA   SPDDAY
00790A B2FF 97 74     A        STAA   LENDAY
00791A B2A1 96 9F     A H9     LDAA   CYCLE
00792A B2A3 80 06     A        SUBA   #006
00793A B2A5 27 5D B302         BEQ    BEGIN
00794                    E
00795A B2A7 96 92     A H3     LDAA   VSDAY1+1
00796A B2A9 B6 74     A        LDAB   VSBUF1+1
00797A B2AB 1B                 ABA                    UPDATE DAY PTR
00798A B2AC 97 92     A        STAA   VSDAY1+1 DAY PTR LSB
00799A B2AE 24 03 B2B3         BCC    H1
00800A B2B0 7C 0091   A        INC    VSDAY1   DAY PTR MSB
00801A B2B3 96 91     A H1     LDAA   VSDAY1
00802A B2B5 B6 73     A        LDAB   VSBUF1   LENGTH OF BUFFER
00803A B2B7 1B                 ABA
00804A B2B8 97 91     A        STAA   VSDAY1
00805A B2BA 96 9E     A        LDAA   PTR+1
00806A B2BC B6 74     A        LDAB   VSBUF1+1
00807A B2BE 1B                 ABA                    UPDATE DATA PTR
00808A B2BF 97 9E     A        STAA   PTR+1    DATA PTR LSB
00809A B2C1 24 03 B2C6         BCC    H2
00810A B2C3 7C 0098   A        INC    PTR      POINTER TO DATA IN BUFFER
00811A B2C6 96 98     A H2     LDAA   PTR
00812A B2C8 B6 73     A        LDAB   VSBUF1
```



```
00813A B2CA 1B                 ABA                    UPDATE DATA PTR
00814A B2CB 97 98     A        STAA   PTR      DATA PTR MSB
00815                    E
00816A B2CD 96 A9     A        LDAA   VSDAYL+1
00817A B2CF B6 74     A        LDAB   VSBUF1+1
00818A B2D1 1B                 ABA
00819A B2D2 97 A9     A        STAA   VSDAYL+1
00820A B2D4 24 03 B2D9         BCC    H11
00821A B2D6 7C 00AB   A        INC    VSDAYL
00822A B2D9 96 AB     A H11    LDAA   VSDAYL
00823A B2DB B6 73     A        LDAB   VSBUF1
00824A B2DD 1B                 ABA
00825A B2DE 97 AB     A        STAA   VSDAYL
00826A B2E0 96 97     A        LDAA   PTRL+1
00827A B2E2 B6 74     A        LDAB   VSBUF1+1
00828A B2E4 1B                 ABA
00829A B2E5 97 97     A        STAA   PTRL+1
00830A B2E7 24 03 B2EC         BCC    H20
00831A B2E9 7C 0096   A        INC    PTRL
00832A B2EC 96 96     A H20    LDAA   PTRL
00833A B2EE B6 73     A        LDAB   VSBUF1
00834A B2F0 1B                 ABA
00835A B2F1 97 96     A        STAA   PTRL
00836                    E
00837A B2F3 96 AB     A        LDAA   VSDAYS+1
00838A B2F5 8B 28     A        ADDA   #028
00839A B2F7 97 AB     A        STAA   VSDAYS+1
00840A B2F9 08                 INX
00841A B2FA 08                 INX
00842A B2FB 08                 INX
00843A B2FC 08                 INX
00844A B2FD 08                 INX
00845A B2FE DF 98     A        STX    PTR8
00846                    E
00847A B300 20 3D B23F         BRA    H4
00848A B302 86 01     A BEGIN  LDAA   #001
00849A B304 97 9F     A        STAA   CYCLE
00850A B306 CE 0400   A        LDX    #START
00851A B309 DF 91     A        STX    VSDAY1   INITIALISE DAY PTR
00852A B30B 08                 INX
00853A B30C 08                 INX
00854A B30D 08                 INX
00855A B30E 08                 INX
00856A B30F DF 98     A        STX    PTR      INITIALISE DATA PTR
00857                    E
00858A B311 BE A4     A        LDX    LENGTH
00859A B313 DF AB     A        STX    VSDAYL
00860A B315 08                 INX
00861A B316 08                 INX
00862A B317 08                 INX
00863A B318 08                 INX
00864A B319 DF 96     A        STX    PTRL
00865                    E
00866A B31B BE A6     A        LDX    SPEED
00867A B31D DF AA     A        STX    VSDAYC
00868A B31F 08                 INX
00869A B320 08                 INX
00870A B321 08                 INX
```

```
PAGE  016  TRY     .BAIO  MEASER

00871A B322 08                  INX
00872A B323 BF 98     A         STX    PTRS
00873                   $
00874A B325 20 10 B33F          BRA    H4
00875                   $
00876A B327 96 9F     A SMALL   LDAA   CYCLE
00877A B329 4C                  INCA
00878A B32A 97 9F     A         STAA   CYCLE
00879A B32C 96 70     A         LDAA   VOLDAY
00880A B32E 80 02     A         SUBA   0002
00881A B330 27 04 B336          BEQ    H10
00882A B332 96 9F     A         LDAA   CYCLE
00883A B334 97 70     A         STAA   VOLDAY
00884A B336 96 9F     A H10     LDAA   CYCLE
00885A B338 80 03     A         SUBA   0003
00886A B33A 27 C6 B302          BEQ    BEGIN
00887A B33C 7E B2A7   A         JMP    H3
00888                   $
00889A B33F BE 91     A H4      LDX    VODAY1
00890A B341 96 03     A         LDAA   YEAR
00891A B343 A7 00     A         STAA   0,X
00892A B345 96 04     A         LDAA   MONTH
00893A B347 A7 01     A         STAA   1,X
00894A B349 96 05     A         LDAA   DAOFMO
00895A B34B A7 02     A         STAA   2,X
00896A B34D 96 09     A         LDAA   DAOFWK
00897A B34F A7 03     A         STAA   3,X
00898                   $
00899A B351 BE A8     A         LDX    VODAYL
00900A B353 96 03     A         LDAA   YEAR
00901A B355 A7 00     A         STAA   0,X
00902A B357 96 04     A         LDAA   MONTH
00903A B359 A7 01     A         STAA   1,X
00904A B35B 96 05     A         LDAA   DAOFMO
00905A B35D A7 02     A         STAA   2,X
00906A B35F 96 09     A         LDAA   DAOFWK
00907A B361 A7 03     A         STAA   3,X
00908                   $
00909A B363 BE AA     A         LDX    VODAYS
00910A B365 96 03     A         LDAA   YEAR
00911A B367 A7 00     A         STAA   0,X
00912A B369 96 04     A         LDAA   MONTH
00913A B36B A7 01     A         STAA   1,X
00914A B36D 96 05     A         LDAA   DAOFMO
00915A B36F A7 02     A         STAA   2,X
00916A B371 96 09     A         LDAA   DAOFWK
00917A B373 A7 03     A         STAA   3,X
00918                   $
00919A B375 39                  RTS
00920                   $
00921                   $         MOTOROLA M6800AM CROSS-ASSEMBLER PAGE 1
00922                   $
00923                   $ M6800AM IS THE PROPERTY OF MOTOROLA SPD, INC.
00924                   $        COPYRIGHT 1974 BY MOTOROLA INC
00925                   $
00926                   $ MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
00927                   $ CALTRANS TO M.T.C MODIFIED, RELEASE 1.0
00928                   $
```

```
00929                   * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
00930                   *
00931                   *          1981 08 08
00932                   *
00933                   *     RELEASE LEVEL 6.75
00934                   *
00935                           MAN    PCS1
00936                   * PCS1. AS   REV. 1.00
00937                   * STEVEN SKELTON
00938                   *
00939                   * FOR TYPE 334 CABINET
00940                   *
00941                   * THIS ROUTINE PROVIDES CORRESPONDENCE BETWEEN HARDWARE DETECTOR
00942                   * ASSIGNMENTS (PORT NO. & BIT NO.) AND THE INTERNAL SOFTWARE
00943                   * DETECTOR ASSIGNMENTS
00944                   *
00945A C2A0                     ORG    INPUT
00946                   *
00947A C2A0 86 D0     A         LDAA   #$D0
00948A C2A2 97 F5     A         STAA   TEMP01   SET UP MSB OF PORT ADDRESSES
00949A C2A4 4F                  CLRA           SET POINTER FOR DETECTOR INPUT PROCESSING AREAS
00950A C2A5 97 EA     A         STAA   INDEX
00951A C2A7 97 EB     A         STAA   INDEX+1
00952                   *
00953                   *        MOTOROLA M68HC11 CROSS-ASSEMBLER PAGE 2
00954                   *
00955A C2A9 97 AC     A         STAA   DATA0H   CLEAR DETECTOR INPUT AREA
00956A C2AB 97 BC     A         STAA   DATA0H+$10
00957A C2AD CE C2BF   A         LDX    #DTRTBL  POINTER TO DETECTOR HARDWARE ASSIGNMENT TABLE
00958A C2B0 DF F8     A         STX    TEMP04
00959A C2B2 C6 01     A IPT1    LDAB   #1       SOFTWARE DETECTOR POSITION MASK
00960                   *
00961A C2B4 37          IPT2    PSHB           SAVE SOFTWARE DETECTOR ID
00962A C2B5 A6 00     A         LDAA   0,X      GET LSB OF PORT ADDRESS
00963A C2B7 97 F4     A         STAA   TEMP02   PUT IT NEXT TO MSB
00964A C2B9 A6 01     A         LDAA   1,X      GET THE HARDWARE DETECTOR POSITION MASK
00965A C2BB DE F5     A         LDX    TEMP01   INPUT PORT ADDRESS IN X
00966A C2BD [illegible] 00    A         ANDA   0,X      SEE IF A CAR OVER DETECTOR
00967A C2BF [illegible] 9 C2C9        BEQ    NXTDET   NO --
00968A C2C1 [illegible] A     A         LDX    INDEX    YES -- GET SOFTWARE DETECTOR PROCESSING INDEX
00969A C2C3 EA AC     A         ORAB   DATA0H,X GET THE INTERNAL DETECTOR BIT
00970A C2C5 E4 C0     A         ANDB   DTCTRH,X KEEP ONLY ENABLED DETECTORS
00971A C2C7 E7 AC     A         STAB   DATA0H,X PUT AWAY THE INTERNAL DETECTOR INPUT WORD
00972                   *
00973A C2C9 DE F8     A NXTDET  LDX    TEMP04   HARDWARE DETECTOR TABLE INDEX
00974A C2CB 08                  INX
00975A C2CC 08                  INX            FIX THE INDEX FOR NEXT ENTRY
00976A C2CD DF F8     A         STX    TEMP04   SAVE THE INDEX
00977A C2CF 33                  PULB           GET THE SOFTWARE DETECTOR POSITON BIT
00978                   *
00979A C2D0 58                  ASLB           SHIFT IT LEFT
00980A C2D1 24 E1 C2B4          BCC    IPT2     NOT DONE WITH THIS GROUP YET
00981A C2D3 79 00EB   A IPT3    TST    INDEX+1  ARE WE ON FIRST GROUP OF DETECTORS?
00982A C2D6 26 06 C2DE          BNE    DONE     NO -- WE'RE DONE
00983A C2D8 86 10     A IPT3    LDAA   #$10     YES -- LETS FIX SOFTWARE DETECTOR AREA INDEX FOR SECOND GROUP
00984A C2DA 97 EB     A         STAA   INDEX+1
00985A C2DC 20 D4 C2B3          BRA    IPT1     DO DO THIS GROUP
00986                   *
```

```
PAGE  018  TOUR    .BAIO  PCB1

009876 C28E 39          DONE   RTS              GO PROCESS THE DETECTORS
00980                   $
00989                   $
00990                   $          MOTOROLA MASSAM CROSS-ASSEMBLER PAGE 2
00991                   $
009928 C28F    01    A  STRTBL FCB    $01       DIRECTION ONE
009938 C2E0    01    A        FCB    TBIT0     LANE ONE
009948 C2E1    01    A        FCB    $01
009958 C2E2    02    A        FCB    TBIT1     LANE TWO
009968 C2E3    01    A        FCB    $01
009978 C2E4    04    A        FCB    TBIT2     LANE THREE
009988 C2E5    01    A        FCB    $01
009998 C2E6    08    A        FCB    TBIT3     LANE FOUR
01000A C2E7    01    A        FCB    $01
01001A C2E8    10    A        FCB    TBIT4     LANE FIVE
01002A C2E9    01    A        FCB    $01
01003A C2EA    20    A        FCB    TBIT5     LANE SIX
01004A C2EB    01    A        FCB    $01
01005A C2EC    40    A        FCB    TBIT6     LANE SEVEN
01006A C2ED    01    A        FCB    $01
01007A C2EE    80    A        FCB    TBIT7     LANE EIGHT
01008                   $
01009A C2EF    03    A        FCB    $03       DIRECTION TWO
01010A C2F0    01    A        FCB    TBIT0     LALE ONE
01011A C2F1    03    A        FCB    $03
01012A C2F2    02    A        FCB    TBIT1     LANE TWO
01013A C2F3    03    A        FCB    $03
01014A C2F4    04    A        FCB    TBIT2     LANE THREE
01015A C2F5    03    A        FCB    $03
01016A C2F6    08    A        FCB    TBIT3     LANE FOUR
01017A C2F7    03    A        FCB    $03
01018A C2F8    10    A        FCB    TBIT4     LANE FIVE
01019A C2F9    03    A        FCB    $03
01020A C2FA    20    A        FCB    TBIT5     LANE SIX
01021A C2FB    03    A        FCB    $03
01022A C2FC    40    A        FCB    TBIT6     LANE SEVEN
01023A C2FD    03    A        FCB    $03
01024A C2FE    80    A        FCB    TBIT7     LANE EIGHT
01025                   $
01026A C2FF    FFFF  A        FDB    $0FFFF
01027A C301    FFFF  A        FDB    $0FFFF
01028A C303    FF    A        FCB    $0FF
01029                   $
01030                   $          MOTOROLA M6800AM CROSS-ASSEMBLER PAGE 1
01031                   $
01032                   $ MASSAM IS THE PROPERTY OF MOTOROLA SPD, INC.
01033                   $        COPYRIGHT 1974 BY MOTOROLA INC
01034                   $
01035                   $ MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
01036                   $ CALTRANS TO M.T.C. MODIFIED, RELEASE 1.0
01037                   $
01038                   $ THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
01039                   $
01040                   $          1981 08 20
01041                   $
01042                   $      RELEASE LEVEL 6.75
01043                   $
01044                          MAM    PCB4
```

```
PAGE  019  TOUR    .BAIO  PCB4

01045                   $
01046                   $ PCB4.AS   REV. 1.00
01047                   $ STEVEN SKELTON
01048                   $
01049                   $  LANE DATA HANDLER ROUTINE
01050                   $
01051                   $ THIS ROUTINE USES THE SIX SECOND VOLUME AND OCCUPANCY
01052                   $ COUNTS TO CALCULATE THE 1 MINUTE RUNNING TOTAL
01053                   $ VOLUME AND OCCUPANCY FOR EACH LANE.
01054                   $
01055                   $ THE ROUTINE PROVIDES A PSUEDO RUNNING TOTAL BY
01056                   $ A.  ONE MINUTE RUNNING TOTAL
01057                   $       1.  PROPORTION CURRENT ONE-MINUTE RUNNING TOTAL TO
01058                   $           9/10THS OF ITSELF.
01059                   $       2.  ADD LATEST 6 SECOND COUNT TO FORM NEW ONE-MINUTE RUNNING TOTAL
01060                   $
01061A C400                    ORG    PCB4
01062                   $
01063A C400 BD C060  A         JSR    SETUP     SETUP INDEX REGISTER AT HIGHEST LANE
01064A C403 BF F8    A         STX    TEMP04    SET UP ADDITIONAL INDEX LOCATION
01065A C405 96 F4    A         LDAA   KAT1      CLEAR LANE VARIABLES?
01066A C407 27 10 C419         BEQ    PROP1     NO -- ON WITH PROGRAM
01067                   $
01068A C409 C6 09    A D1      LDAB   #11       VARIABLES TO CLEAR PER LANE
01069A C40B 6F 00    A D2      CLR    0,IX1,X
01070A C40D 08                 INX
01071A C40E 5A                 DECB             ALL VARIABLES CLEARED?
01072A C40F 26 FA C40B         BNE    D2        IF NOT, GET ANOTHER
01073A C411 BD C030  A         JSR    NEXTLN    POINT TO NEXT LANE'S VARIABLES
01074A C414 24 F3 C409         BCC    D1        CLEAR THEM IF CARRY CLEAR
01075                   $
01076A C416 BD C060  A         JSR    SETUP     SET MAIN LANE DATA INDEXES
01077                   $
01078A C419 96 F3    A PROP1   LDAA   PCN1      1 MIN. ART PROPORTION CONTROL
01079A C41B C6 02    A         LDAB   #0CNN1H    SET TO GET 1 MIN OCCUPANCY
01080A C41D BD 83 C472         BSR    PRO16     PROPORTION IT
01081A C41F 96 F3    A         LDAA   PCN1
01082A C421 C6 04    A         LDAB   #VNN1      SET TO GET 1 MIN VOLUME
01083A C423 BD 3F C464         BSR    PRO8      PROPORTION IT
01084                   $
01085                   $
01086                   $          MOTOROLA M6800AM CROSS-ASSEMBLER PAGE 3
01087                   $
01088A C425 BD C030  A         JSR    NEXTLN    LAST LANE DONE?
01089A C428 24 EF C419         BCC    PROP1     NO -- DO NEXT ONE
01090                   $
01091A C42A BD C060  A ABTBL   JSR    SETUP     POINT INDEX REGISTER TO HIGHEST LANE
01092A C42D A6 00    A D50     LDAA   0,IX1,X   SIX SECOND OCCUPANCY
01093A C42F 16                 TAB              COPY OCCUPANCY INTO B
01094A C430 EB 06    A         ADDB   0CC1L,X   ADD TO 30 SECOND OCCUPANCY - LSB
01095A C432 E7 06    A         STAB   0CC1L,X   & SAVE IT
01096A C434 24 02 C438         BCC    D3        OVERFLOW?
01097A C436 6C 05    A         INC    0CC1H,X   YES, INCREMENT MSB
01098A C438 16          D3     TAB              COPY OCCUPANCY INTO B, AGAIN
01099A C439 AB 03    A         ADDA   0CNN1L,X  ADD TO 1 MINUTE OCCUPANCY - LSB
01100A C43B A7 03    A         STAA   0CNN1L,X  & SAVE IT
01101A C43D 24 02 C441         BCC    D5        OVERFLOW?
01102A C43F 6C 02    A         INC    0CNN1H,X  YES, INCREMENT MSB
```

```
PAGE  020  TSUR    .BAID  PCB4

01103A C041 6F 00     A 3S      CLR     0BIX1,X   CLEAR A SEC OCCUPANCY
01104A C043 A6 01     A         LDAA    VBIX1,X   SIX SECOND VOLUME
01105A C045 16                  TAB               COPY VOLUME INTO B
01106A C046 EB 07     A         ADDB    VOL1,X    ADD TO 30 SECOND VOLUME
01107A C048 E7 07     A         STAB    VOL1,X    & SAVE IT
01108A C04A 16                  TAB               COPY VOLUME INTO B, AGAIN
01109A C04B EB 09     A         ADDB    VOIL,X    ADD A SEC.  VOL TO 1-MN VOL
01110A C04D E7 09     A         STAB    VOIL,X    & SAVE IT
01111A C04F 24 02 C053          BCC     BS4       OVERFLOW?
01112A C051 6C 08     A         INC     VOIN,X    YES, INCREMENT MSB
01113A C053 48          BS4     ASLA              DOUBLE THE VOLUMES
01114A C054 16                  TAB               COPY VOLUME INTO B, AGAIN
01115A C055 AB 04     A         ADDA    VNM1,X    ADD TO 1 MINUTE VOLUME
01116A C057 A7 04     A         STAA    VNM1,X    & SAVE IT
01117A C059 6F 01     A         CLR     VBIX1,X   CLEAR A SEC VOLUME
01118A C05B BD C030   A         JSR     NEXTLN    GET NEXT LANE'S ADDRESS
01119                   0
01120                   0         MOTOROLA MAGSAM CROSS-ASSEMBLER PAGE 4
01121                   0
01122A C05E 24 CD C02D          BCC     BS0       GET ANOTHER LANE OR
01123A C060 7F 00F4   A         CLR     KAY1      RESET THE LANE DATA TABLE FLAG
01124A C063 39                  RTS               RETURN TO CALLER
01125                   0
01126A C064 DD E3     A PRCB    ADDD    INDEX+1   COMPUTE ADDRESS OF DATA
01127A C066 B7 F9     A         STAB    TEMP05    STORE IT
01128A C068 BE F8     A         LDX     TEMP04    GET X REG
01129A C06A BD C080   A         JSR     PRPRTB    DO THE PROPORTIONING
01130A C06D BE F8     A         LDX     TEMP04    RESET X REG
01131A C06F E7 00     A         STAB    0,X       PUT AWAY NEW DATA
01132A C071 39                  RTS
01133                   0
01134A C072 DD E3     A PRS16   ADDD    INDEX+1
01135A C074 B7 F9     A         STAB    TEMP05
01136A C076 BE F8     A         LDX     TEMP04
01137A C078 BD C080   A         JSR     PRPRT     DO A 2-BYTE PROPORTION
01138A C07B BE F8     A         LDX     TEMP04
01139A C07D E7 01     A         STAB    1,X       SALT AWAY LSB
01140A C07F A7 00     A         STAA    0,X       SAVE MSB
01141A C081 39                  RTS
01142A C082     FFFF  A         FDB     $0FFFF
01143                   0
01144                   0         MOTOROLA MAGSAM CROSS-ASSEMBLER PAGE 1
01145                   0
01146                   0 MAGSAM IS THE PROPERTY OF MOTOROLA EPD, INC.
01147                   0         COPYRIGHT 1974 BY MOTOROLA INC
01148                   0
01149                   0 MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
01150                   0 CALTRANS TO N.T.C. MODIFIED, RELEASE 1.0
01151                   0
01152                   0 THIS PROGRAM MATERIAL FOR DESIGN STUDY ...?
01153                   0
01154                   0         1981 08 20
01155                   0
01156                   0       RELEASE LEVEL 6.73
01157                   0
01158                             NAM     PCB4A
01159                   0
01160                   0 PCB4A.AS   REV. 1.50
```

```
PAGE  021  TSUR    .BAID  PCB4A

01161                   0 STEVEN SKELTON
01162                   0
01163                   0  LANE DATA HANDLER ROUTINE
01164                   0
01165                   0 THIS ROUTINE USES THE SIX SECOND VOLUME AND OCCUPANCY
01166                   0 COUNTS TO CALCULATE THE 1 MINUTE RUNNING TOTAL
01167                   0 VOLUME AND OCCUPANCY FOR EACH LANE.
01168                   0
01169                   0 THE ROUTINE PROVIDES A PSUEDO RUNNING TOTAL BY
01170                   0 A.  ONE MINUTE RUNNING TOTAL
01171                   0       1.  PROPORTION CURRENT ONE-MINUTE RUNNING TOTAL TO
01172                   0           9/10THS OF ITSELF.
01173                   0       2.  ADD LATEST 6 SECOND COUNT TO FORM NEW ONE-MINUTE RUNNING TOTAL
01174                   0
01175A C040                     ORG     PCB4A
01176                   0
01177A C040 BD C060   A         JSR     SETUP     SETUP INDEX REGISTER AT HIGHEST LANE
01178A C043 DF F8     A         STX     TEMP04    SET UP ADDITIONAL INDEX LOCATION
01179A C045 96 A2     A         LDAA    KAY2      CLEAR LANE VARIABLES?
01180A C047 27 10 C059          BEQ     PGP1      NO -- GO WITH PROGRAM
01181                   0
01182A C049 C6 0B     A DA1     LDAB    011       VARIABLES TO CLEAR PER LANE
01183A C04B 6F 00     A DA2     CLR     0BIX1+000,X
01184A C04D 08                  INX
01185A C04E 5A                  DECB              ALL VARIABLES CLEARED?
01186A C04F 26 FA C04B          BNE     DA2       IF NOT, GET ANOTHER
01187A C051 BD C030   A         JSR     NEXTLN    POINT TO NEXT LANE'S VARIABLES
01188A C054 24 F3 C049          BCC     DA1       CLEAR THEM IF CARRY CLEAR
01189                   0
01190A C056 BD C060   A         JSR     SETUP     SET MAIN LANE DATA INDEXES
01191                   0
01192A C059 96 F3     A PGP1    LDAA    PCW1      1 MIN. ART PROPORTION CONTROL
01193A C05B C6 02     A         LDAB    #OCNM1H+000 SET TO GET 1 MIN OCCUPANCY
01194A C05D BD C072   A         JSR     PRS16     PROPORTION IT
01195A C060 96 F3     A         LDAA    PCW1
01196A C062 C6 04     A         LDAB    #VNM1+000 SET TO GET 1 MIN VOLUME
01197A C064 BD C064   A         JSR     PRCB      PROPORTION IT
01198                   0
01199                   0
01200                   0         MOTOROLA MAGSAM CROSS-ASSEMBLER PAGE 3
01201                   0
01202A C067 BD C030   A         JSR     NEXTLN    LAST LANE DONE?
01203A C06A 24 E9 C059          BCC     PGP1      NO -- DO NEXT ONE
01204                   0
01205A C06C BD C060   A         JSR     SETUP     POINT INDEX REGISTER TO HIGHEST LANE
01206A C06F A6 00     A DA30    LDAA    0BIX1+000,X SIX SECOND OCCUPANCY
01207A C071 16                  TAB               COPY OCCUPANCY INTO B
01208A C072 EB 06     A         ADDB    OCC1L+000,X ADD TO 30 SECOND OCCUPANCY - LSB
01209A C074 E7 06     A         STAB    OCC1L+000,X & SAVE IT
01210A C076 24 02 C07A          BCC     DA3       OVERFLOW?
01211A C078 6C 05     A         INC     OCC1M+000,X YES, INCREMENT MSB
01212A C07A 16          DA3     TAB               COPY OCCUPANCY INTO B, AGAIN
01213A C07B AB 03     A         ADDA    OCNM1L+000,X ADD TO 1 MINUTE OCCUPANCY - LSB
01214A C07D A7 03     A         STAA    OCNM1L+000,X & SAVE IT
01215A C07F 24 02 C083          BCC     DA5       OVERFLOW?
01216A C081 6C 02     A         INC     OCNM1H+000,X YES, INCREMENT MSB
01217A C083 6F 00     A DA5     CLR     0BIX1+000,X CLEAR A SEC OCCUPANCY
01218A C085 A6 01     A         LDAA    VBIX1+000,X SIX SECOND VOLUME
```

```
PAGE  022  YOUR    .SAIO  PCS4A

01219A C887 16                  TAB             COPY VOLUME INTO B
01220A C888 EB 87     A         ADDB    VOL1+$80,X ADD TO 30 SECOND VOLUME
01221A C88A E7 87     A         STAB    VOL1+$80,X & SAVE IT
01222A C88C 16                  TAB             COPY VOLUME INTO B, AGAIN
01223A C88D EB 89     A         ADDB    VD1L+$80,X ADD 6 SEC.  VOL TO 1-HR VOL
01224A C88F E7 89     A         STAB    VD1L+$80,X & SAVE IT
01225A C891 24 02 C895          BCC     DA56    OVERFLOW?
01226A C893 6C 88     A         INC     VD1M+$80,X YES, INCREMENT MSB
01227A C895 48          DA56    ASLA            DOUBLE THE VOLUMES
01228A C896 16                  TAB             COPY VOLUME INTO B, AGAIN
01229A C897 AB 84     A         ADDA    VMN1+$80,X ADD TO 1 MINUTE VOLUME
01230A C899 A7 84     A         STAA    VMN1+$80,X & SAVE IT
01231A C89B 6F 81     A         CLR     VSIX1+$80,X CLEAR 6 SEC VOLUME
01232A C89D BB C030   A         JBR     NEXTLN  GET NEXT LANE'S ADDRESS
01233                   *
01234                   *         MOTOROLA M68BAM CROSS-ASSMBLER PAGE 4
01235                   *
01236A C8A0 24 CD C86F          BCC     DA50    GET ANOTHER LANE OR
01237A C8A2 7F 00A2   A         CLR     KAY2    RESET THE LANE DATA TABLE FLAG
01238A C8A5 39                  RTS             RETURN TO CALLER
01239                   *
01240A C8A6     FFFF  A         FDB     $0FFFF
01241                   *
01242                   *         MOTOROLA M68BAM CROSS-ASSMBLER PAGE 1
01243                   *
01244                   * M68BAM IS THE PROPERTY OF MOTROLA GPD, INC.
01245                   *         COPYRIGHT 1974 BY MOTOROLA INC
01246                   *
01247                   * MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
01248                   * CALTRANS TO M.T.C. MODIFIED, RELEASE 1.0
01249                   *
01250                   * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
01251                   *
01252                   *             1981 08 20
01253                   *
01254                   *       RELEASE LEVEL 6.75
01255                   *
01256                           NAM     PCS5
01257                   *
01258                   * PCS5.AG  REV 1.00
01259                   * STEVEN SKELTON
01260                   *
01261                   * THIS ROUTINE SUMS THE RUNNING TOTALS FOR UP TO EIGHT VALID LANES
01262                   * AND DIVIDES BY THE NUMBER OF VALID LANES, PROVIDING
01263                   * AN AVERAGE RUNNING TOTAL.
01264                   *
01265                   *
01266                   * FOUR SEPARATE AVERAGE RUNNING TOTALS (ART) ARE
01267                   * MAINTAINED. ONE FOR VOLUME AND ONE FOR OCCUPANCY
01268                   * FOR  1 MINUTE DATA.
01269                   *
01270A C4A0                     ORG     PCS5
01271                   *
01272                   *           MOTOROLA M68BAM CROSS-ASSMBLER PAGE 2
01273                   *
01274A C4A0 C6 04     A         LDAB    $04     4 VARIABLES TO CLEAR
01275A C4A2 CE 00EC   A         LDX     $ARIMOM
01276A C4A5 6F 00     A LOOP1   CLR     0,X     CLEAR ART
```

```
PAGE  023  YOUR    .BA10  PC85

C4A7 08                  INX              NEXT ART
C4A8 5A                  DECD             LAST ONE CLEARED?
C4A9 26 FA C4A5          BNE    LOOP1     IF NOT, CLEAR ANOTHER.
                  *
C4AB C6 08     A         LDAB   08        COUNTER FOR VALID LANES
C4AD BD C060   A         JSR    SETUP     POINT TO LANE 5 AREA
C4B0 A6 0A     A LOOP2   LDAA   DET1,X    DETECTOR ERROR WORD
C4B2 48                  ASLA             DETECTOR ERROR BIT TO CARRY
C4B3 24 06 C4BB          BCC    E5        IF DATA O.K. - GO ON
C4B5 5A           E3     DECB             DECREMENT VALID LANE COUNT
C4B6 44                  LSRA             CLEAR DETECTOR ERROR BIT
C4B7 A7 0A     A         STAA   DET1,X    RESTORE ERROR WORD
C4B9 20 21 C4BC          BRA    NLOOP2    SEE IF ALL LANES COUNTED
                  *
C4BB 6D 0C     A E5      TST    LNDIS1,X  DISABLE THIS LANE'S DATA?
C4BD 26 F6 C4B5          BNE    E3        YES -- DECREMENT VALID LANE COUNT & SET FOR NEXT LANE
C4BF A6 04     A         LDAA   VMN1,X    1 MINUTE VOLUME
C4C1 36                  PSHA
C4C2 A6 02     A         LDAA   OCMN1M,X  1 MINUTE OCCUPANCY - MSB
C4C4 36                  PSHA
C4C5 A6 03     A         LDAA   OCMN1L,X  1 MINUTE OCCUPANCY - LSB
C4C7 CE 00EC   A         LDX    #AR1MOM   POINT TO ART TABLE
C4CA AB 01     A LOOP3   ADDA   1,X       ADD TO OCCUPANCY ART - LSB
C4CC A7 01     A         STAA   1,X       STORE
C4CE 32                  PULA             OCCUPANCY - MSB
C4CF A9 00     A         ADCA   0,X       ADD TO OCCUPANCY ART - MSB
C4D1 A7 00     A         STAA   0,X       STORE
C4D3 32                  PULA             VOLUME
                  *
                  *          MOTOROLA M68SAM CROSS-ASSEMBLER PAGE 3
                  *
C4D4 AB 03     A         ADDA   3,X       ADD TO VOLUME ART - LSB
C4D6 A7 03     A         STAA   3,X       STORE
C4D8 24 02 C4DC          BCC    NLOOP2    IF NO CARRY, LEAVE MSB ALONE
C4DA 6C 02     A         INC    2,X       ADD CARRY TO MSB
                  *
                  *
C4DC BD C030   A NLOOP2  JSR    NEXTLN    GET NEXT LANE'S AREA
C4DF 24 CF C4B0          BCC    LOOP2     IF ALL LANES NOT ADDED, GO GET ANOTHER
                  *
C4E1 CE C500   A AVG     LDX    #PCWTAB   SET INDEX REGISTER
C4E4 5C                  INCD
C4E5 5A           LOOP4  DECD             DECREMENT LANE COUNT
C4E6 27 03 C4EB          BEQ    PIKUP     STOP LOOP WHEN POINTER CORRECTLY POSITIONED
C4E8 08                  INX              MOVE POINTER TO NEXT PCW
C4E9 20 [illegible] C4E5 BRA    LOOP4     CHECK ON POINTER POSITION AGAIN
                  *
C4EB A6 00     A PIKUP   LDAA   0,X       PROPER PROPORTION CONTROL WORD(PCW)
C4ED CE 00ED   A         LDX    #AR1MOM   POINTER TO SUMMED RUNNING TOTALS
C4F0 36           LOOP5  PSHA             SAVE PCW ON STACK
C4F1 DF EA     A         STX    INDEX     SAVE INDEX REGISTER
C4F3 BD C0B0   A         JSR    PRPRT     PROPORTION THE LANE'S SUM
C4F6 DE EA     A         LDX    INDEX     POINT TO ART TABLE
C4F8 A7 00     A         STAA   0,X       STORE MSB
C4FA E7 01     A         STAB   1,X       STORE LSB
C4FC 8C 00EC   A         CPX    #AR1MOL-1 FINISHED ALL PROPORTIONING?
C4FF 27 05 C506          BEQ    OUT       IF SO, CLEAN UP STACK AND BACK TO CALLER
C501 08                  INX              POINT TO NEXT
```

```
PAGE  024  YOUR    .SA:0  PCS5

01335A C502 08                  INX              LANE SUM
01336A C503 32                  PULA             PCW FROM STACK
01337                     *
01338                     *          MOTOROLA M68SAM CROSS-ASSMBLER PAGE 4
01339                     *
01340A C504 20 EA C4F0          BRA    LOOP5     BACK FOR NEXT ART
01341                     *
01342A C506 31            OUT   INS              DELETE PCW FROM STACK
01343A C507 39            OUT1  RTS              BACK TO CALLER
01344A C508    FF     A PCWTAB FCB    $0FF      PCW FOR 0 LANES
01345A C509    00     A        FCB    PCLN1     STORE PROPORTION CONTROL WORDS
01346A C50A    80     A        FCB    PCLN2     IN PROM FOR AVERAGING THE OCCUPANCIES
01347A C50B    AB     A        FCB    PCLN3     AND VOLUMES FOR THE NUMBER OF VALID LANES
01348A C50C    C0     A        FCB    PCLN4     AT THIS SITE.
01349A C50D    CD     A        FCB    PCLN5
01350A C50E    D5     A        FCB    PCLN6
01351A C50F    DB     A        FCB    PCLN7
01352A C510    E0     A        FCB    PCLN8
01353                     *
01354                     *        MOTOROLA M68SAM CROSS-ASSMBLER PAGE 1
01355                     *
01356                     * M68SAM IS THE PROPERTY OF MOTROLA SPD, INC.
01357                     *        COPYRIGHT 1974 BY MOTOROLA INC
01358                     *
01359                     * MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
01360                     * CALTRANS TO M.T.C. MODIFIED, RELEASE 1.0
01361                     *
01362                     * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
01363                     *
01364                     *            1981 08 20
01365                     *
01366                     *      RELEASE LEVEL 6.75
01367                     *
01368                           NAM    PCS5A
01369                     *
01370                     * PCS5A.AS  REV 1.00
01371                     * STEVEN SKELTON
01372                     *
01373                     * THIS ROUTINE SUMS THE RUNNING TOTALS FOR UP TO EIGHT VALID LANES
01374                     * AND DIVIDES BY THE NUMBER OF VALID LANES, PROVIDING
01375                     * AN AVERAGE RUNNING TOTAL.
01376                     *
01377                     *
01378                     * TWO SEPARATE AVERAGE RUNNING TOTALS (ART) ARE
01379                     * MAINTAINED. ONE FOR VOLUME AND ONE FOR OCCUPANCY
01380                     * FOR  1 MINUTE DATA.
01381                     *
01382A C8D0                     ORG    PCS5A
01383                     *
01384                     *          MOTOROLA M68SAM CROSS-ASSMBLER PAGE 2
01385                     *
01386A C8D0 C6 04     A         LDAB   $04       4 VARIABLES TO CLEAR
01387A C8D2 CE 00FC   A         LDX    $AR1MOM+$10
01388A C8D5 6F 00     A LOOPA1  CLR    0,X       CLEAR ART
01389A C8D7 08                  INX              NEXT ART
01390A C8D8 5A                  DECB             LAST ONE CLEARED?
01391A C8D9 26 FA C8D5          BNE    LOOPA1    IF NOT, CLEAR ANOTHER.
01392                     *
```

```
PAGE  025  YOUR    .SA:0  PCS5A

C8DD C6 08     A        LDAB   #8       COUNTER FOR VALID LANES
C8DD BD C060   A        JSR    SETUP    POINT TO LANE 5 AREA
C8E0 A6 8A     A LOOPA2 LDAA   DET1+$80,X DETECTOR ERROR WORD
C8E2 48                 ASLA            DETECTOR ERROR BIT TO CARRY
C8E3 24 06 C8EB         BCC    EA5      IF DATA O.K. - GO ON
C8E5 5A          EA3    DECB            DECREMENT VALID LANE COUNT
C8E6 44                 LSRA            CLEAR DETECTOR ERROR BIT
C8E7 A7 8A     A        STAA   DET1+$80,X RESTORE ERROR WORD
C8E9 20 21 C90C         BRA    NOOP2    SEE IF ALL LANES COUNTED
                 $
C8EB 6B 8C     A EA5    TST    LNDIS1+$80,X DISABLE THIS LANE'S DATA?
C8ED 26 F6 C8E5         BNE    EA3      YES -- DECREMENT VALID LANE COUNT & SET FOR NEXT LANE
C8EF A6 84     A        LDAA   VMN1+$80,X 1 MINUTE VOLUME
C8F1 36                 PSHA
C8F2 A6 82     A        LDAA   OCMN1M+$80,X 1 MINUTE OCCUPANCY - MSB
C8F4 36                 PSHA
C8F5 A6 83     A        LDAA   OCMN1L+$80,X 1 MINUTE OCCUPANCY - LSB
C8F7 CE 00FC   A        LDX    #AR1MOM+$10 POINT TO ART TABLE
C8FA AB 01     A LOOPA3 ADDA   1,X      ADD TO OCCUPANCY ART - LSB
C8FC A7 01     A        STAA   1,X      STORE
C8FE 32                 PULA            OCCUPANCY - MSB
C8FF A9 00     A        ADCA   0,X      ADD TO OCCUPANCY ART - MSB
C901 A7 00     A        STAA   0,X      STORE
C903 32                 PULA            VOLUME
                 $
                 $          MOTOROLA M68SAM CROSS-ASSMBLER PAGE 3
                 $
C904 AB 03     A        ADDA   3,X      ADD TO VOLUME ART - LSB
C906 A7 03     A        STAA   3,X      STORE
C908 24 02 C90C         BCC    NOOP2    IF NO CARRY, LEAVE MSB ALONE
C90A 6C 02     A        INC    2,X      ADD CARRY TO MSB
                 $
                 $
C90C BD C030   A NOOP2  JSR    NEXTLN   GET NEXT LANE'S AREA
C90F 24 CF C8E0         BCC    LOOPA2   IF ALL LANES NOT ADDED, GO GET ANOTHER
                 $
C911 CE C508   A        LDX    #PCWTAB  SET INDEX REGISTER
C914 5C                 INCB
C915 5A          LOOPA4 DECB            DECREMENT LANE COUNT
C916 27 03 C91B         BEQ    PICKUP   STOP LOOP WHEN POINTER CORRECTLY POSITIONED
C918 08                 INX             MOVE POINTER TO NEXT PCW
C919 20 FA C915         BRA    LOOPA4   CHECK ON POINTER POSITION AGAIN
                 $
C91B A6 00     A PICKUP LDAA   0,X      PROPER PROPORTION CONTROL WORD(PCW)
C91D CE 00FC   A        LDX    #AR1MOM+$10 POINTER TO SUMMED RUNNING TOTALS
C920 36          LOOPA5 PSHA            SAVE PCW ON STACK
C921 DF EA     A        STX    INDEX    SAVE INDEX REGISTER
C923 BD C0B0   A        JSR    PRPRT    PROPORTION THE LANE'S SUM
C926 DE EA     A        LDX    INDEX    POINT TO ART TABLE
C928 A7 00     A        STAA   0,X      STORE MSB
C92A E7 01     A        STAB   1,X      STORE LSB
C92C 8C 00FE   A        CPX    #AR1MVL-1+$10 FINISHED ALL PROPORTIONING?
C92F 27 05 C936         BEQ    OUTT     IF SO, CLEAN UP STACK AND BACK TO CALLER
C931 08                 INX             POINT TO NEXT
C932 08                 INX             LANE SUM
C933 32                 PULA            PCW FROM STACK
                 $
                 $          MOTOROLA M68SAM CROSS-ASSMBLER PAGE 4
```

```
01481                          *
01482A C934 20 EA C920               BRA    LOOPAS    BACK FOR NEXT ART
01483                          *
01484A C936 31                 OUTT   INS              DELETE PCW FROM STACK
01485A C937 39                 OUTT1  RTS              BACK TO CALLER
01486                          *          MOTOROLA M68BAM CROSS-ASSEMBLER PAGE 1
01487                          *
01488                          * M68BAM IS THE PROPERTY OF MOTOROLA SPD, INC.
01489                          *        COPYRIGHT 1974 BY MOTOROLA INC
01490                          *
01491                          * MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
01492                          * CALTRANS TO M.T.C. MODIFIED, RELEASE 1.0
01493                          *
01494                          * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
01495                          *
01496                          *            1981 08 24
01497                          *
01498                          *      RELEASE LEVEL 6.75
01499                          *
01500                                 NAM    PCS6
01501                          * PCS6.AS REV. 1.00
01502                          * STEVEN SKELTON
01503                          *
01504                          * THIS ROUTINE MONITORS MAIN LANE DETECTORS
01505                          *  FOR FAILURE EVERY 4 SEC.
01506                          *
01507                          *  1.  OCCUPANCY TEST #1 (OCTST1)
01508                          *      PERFORMED - ALWAYS
01509                          *      FAILURE - IF 1 MIN OCCUPANCY OF LANE > 'OC10CN,L'
01510                          *                - SETS BIT 2 IN DETECTOR STATUS TABLE
01511                          *
01512                          * ANY FAILURE SETS NS BIT OF THAT LANE'S DETECTOR ERROR WORD. (DETX)
01513                          *
01514A C550                           ORG    PCS6
01515                          *
01516          0002  A 01BIT  EQU    $02
01517          0080  A ERRBIT EQU    $80
01518                          *
01519A C550 CE 0201  A        LDX    #OCNN1N$256+1 PTR FOR 1 MIN OCCUPANCY
01520A C553 DF F8    A        STX    TEMP04
01521A C555 01                NOP
01522A C556 01                NOP
01523                          *
01524A C557 DE F8    A OCTST1 LDX    OC10CN   GET OCCUPANCY LIMIT
01525A C559 DF F6    A        STX    TEMP02   STORE FOR OCTEST
01526A C55B BD C060  A        JSR    SETUP    SET DATA INDEXES FOR M.L. DATA
01527A C55E BD 42 C5A2 L01    BSR    OCTEST   IS LANE OCC > LIMIT?
01528A C560 24 04 C566        BCC    L07      NO --
01529A C562 86 02    A        LDAA   #01BIT   YES -- ERROR BIT
01530A C564 8D 33 C599        BSR    ERSET    PROCESS ERROR FLAGS
01531A C566 E6 0A    A L07    LDAB   DET1,X   GET CURRENT DETECTOR ERROR WORD
01532A C568 58                ASLB
01533                          *
01534                          *          MOTOROLA M68BAM CROSS-ASSEMBLER PAGE 3
01535                          *
01536A C569 58                ASLB           ACTIVE STATE CHANGE ERROR?
01537A C56A 24 04 C570        BCC    L09      NO -- GO ON
01538A C56C 86 80    A        LDAA   #ERRBIT  YES -- FLAG THIS DETECTOR
```



```
01539A C56E 8D 29 C599        BSR    ERSET    YES -- FLAG THIS DETECTOR
01540A C570 BD C030  A L09    JSR    NEXTLN   LAST LANE?
01541A C573 24 E9 C55E        BCC    L01      NO -- LOOP BACK
01542                          *
01543A C575 C6 01    A        LDAB   #$01     SET O/P LANE PTR
01544A C577 CE 0100  A        LDX    #$0100   SET DETECTOR PTR
01545A C57A B7 F7    A        STAB   TEMP03   STORE O/P LANE POINTER
01546A C57C A6 0A    A L012   LDAA   DET1,X   GET ERROR BYTE
01547A C57E 48                ASLA           ERROR BIT SET?
01548A C57F 24 04 C585        BCC    L011     NO
01549A C581 BA 88    A        ORAB   DETER1   SET ALARM
01550A C583 B7 88    A        STAB   DETER1   STORE ALARM
01551A C585 B6 F7    A L011   LDAB   TEMP03
01552A C587 58                ASLB           INCREMENT O/P LANE PTR
01553A C588 B7 F7    A        STAB   TEMP03   STORE PTR
01554A C58A 25 0C C598        BCS    L013     ALL DONE?
01555A C58C DF F5    A        STX    TEMP01   STORE DETECTOR PTR
01556A C58E 96 F6    A        LDAA   TEMP02
01557A C590 8B 10    A        ADDA   #$10     UPDATE DETECTOR PTR
01558A C592 97 F6    A        STAA   TEMP02   STORE IT!
01559A C594 DE F5    A        LDX    TEMP01   LOAD DETECTOR PTR
01560A C596 20 E4 C57C        BRA    L012
01561A C598 39                L013   RTS
01562                          *
01563                          *
01564A C599 E6 0C    A ERSET  LDAB   LNDIS1,X LANE DISABLED?
01565A C59B 26 04 C5A1        BNE    ER9      YES -- GO BACK
01566A C59D AA 0A    A        ORAA   DET1,X   SET BIT
01567A C59F A7 0A    A        STAA   DET1,X   STORE IT
01568A C5A1 39                ER9    RTS      BACK TO CALLER
01569                          *
01570A C5A2 96 EB    A OCTEST LDAA   INDEX+1
01571A C5A4 9B F8    A        ADDA   TEMP04
01572A C5A6 97 FA    A        STAA   TEMP06
01573A C5A8 DE F9    A        LDX    TEMP05
01574A C5AA BD C000  A        JSR    CMP16    MAKE OCCUPANCY COMPARISON
01575A C5AD DE EA    A        LDX    INDEX    RESTORE X
01576A C5AF 39                RTS
01577                          *
01578                          *          MOTOROLA M68BAM CROSS-ASSEMBLER PAGE 1
01579                          *
01580                          * M68BAM IS THE PROPERTY OF MOTOROLA SPD, INC.
01581                          *        COPYRIGHT 1974 BY MOTOROLA INC
01582                          *
01583                          * MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
01584                          * CALTRANS TO M.T.C. MODIFIED, RELEASE 1.0
01585                          *
01586                          * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
01587                          *
01588                          *            1981 08 24
01589                          *
01590                          *      RELEASE LEVEL 6.75
01591                          *
01592                                 NAM    PCS6A
01593                          * PCS6A.AS REV. 1.00
01594                          * STEVEN SKELTON
01595                          *
01596                          * THIS ROUTINE MONITORS MAIN LANE DETECTORS
```

```
PAGE  028  YOUR    .SA10  PCS6A

                         *  FOR FAILURE EVERY 6 SEC.
                         *
                         *  1.  OCCUPANCY TEST #1 (OCTST1)
                         *      PERFORMED - ALWAYS
                         *      FAILURE - IF 1 MIN OCCUPANCY OF LANE > "OC1OCM,L"
                         *              - SETS BIT 2 IN DETECTOR STATUS TABLE
                         *
                         * ANY FAILURE SETS MS BIT OF THAT LANE'S DETECTOR ERROR WORD. (DETX)
                         *
A C950                          ORG    PCS6A
                         *
                         *
A C950 CE 8201   A              LDX    #OCMN1M*256+*80*256+1 PTR FOR 1 MIN OCCUPANCY
A C953 DF F8     A              STX    TEMP04
A C955 01                       NOP
A C956 01                       NOP
                         *
A C957 DE 9B     A              LDX    OC1OCM   GET OCCUPANCY LIMIT
A C959 DF F6     A              STX    TEMP02   STORE FOR OCTEST
A C95B BD C060   A              JSR    SETUP    SET DATA INDEXES FOR M.L. DATA
A C95E BD C5A2   A LOA1         JSR    OCTEST   IS LANE OCC > LIMIT?
A C961 24 05 C968               BCC    LOA7     NO --
A C963 86 82     A              LDAA   #01BIT   YES -- ERROR BIT
A C965 BD C99C   A              JSR    ERSETT   PROCESS ERROR FLAGS
A C968 E6 8A     A LOA7         LDAB   DET1+*80,X GET CURRENT DETECTOR ERROR WORD
A C96A 58                       ASLB
                         *
                         *          MOTOROLA M68SAM CROSS-ASSMBLER PAGE 3
                         *
A C96B 58                       ASLB             ACTIVE STATE CHANGE ERROR?
A C96C 24 05 C973               BCC    LOA9     NO -- GO ON
A C96E 86 80     A              LDAA   #ERRBIT  YES -- FLAG THIS DETECTOR
A C970 BD C99C   A              JSR    ERSETT
A C973 BD C030   A LOA9         JSR    NEXTLN   LAST LANE?
A C976 24 E6 C95E               BCC    LOA1     NO  -- LOOP BACK
                         *
A C978 C6 01     A              LDAB   #$01     SET O/P LANE PTR
A C97A CE 0100   A              LDX    #$100    SET DETECTOR PTR
A C97D D7 F7     A              STAB   TEMP03   STORE O/P LANE PTR
A C97F A6 8A     A LOA12        LDAA   DET1+*80,X GET ERROR BIT
A C981 48                       ASLA             ERROR BIT SET?
A C982 24 04 C988               BCC    LOA11    NO
A C984 DA 8A     A              ORAB   DETER3   SET ALARM
A C986 D7 8A     A              STAB   DETER3   STORE ALARM
A C988 D6 F7     A LOA11        LDAB   TEMP03
A C98A 58                       ASLB             INCREMENT O/P LANE PTR
A C98B D7 F7     A              STAB   TEMP03   STORE PTR
A C98D 25 0C C99B               BCS    LOA13    ALL DONE?
A C98F DF F5     A              STX    TEMP01   STORE DETECTOR PTR
A C991 96 F6     A              LDAA   TEMP02
A C993 8B 10     A              ADDA   #$10     UPDATE DETECTOR PTR
A C995 97 F6     A              STAA   TEMP02   STORE IT!
A C997 DE F5     A              LDX    TEMP01   LOAD DETECTOR PTR
A C999 20 E4 C97F               BRA    LOA12
A C99B 39        LOA13          RTS
                         *
                         *
A C99C E6 8C     A ERSETT       LDAB   LNDIS1+*80,X
```

```
01625A C9FE 26 04 C9A4          BNE    ER90
01626A C9A0 AA 8A     A         ORAA   DET1+080,X
01627A C9A2 A7 8A     A         STAA   DET1+080,X
01628A C9A4 39                  RTS
01629                   ER90    *
01630                           *
01631                           *        MOTOROLA M6800 CROSS-ASSEMBLER PAGE 1
01632                           *
01633                           * M6800 IS THE PROPERTY OF MOTOROLA SPD, INC.
01634                           * COPYRIGHT 1974 BY MOTOROLA INC
01635                           *
01636                           * MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
01637                           * WRITTEN BY M.T.C. , RELEASE 2.0
01638                           *
01639                           * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
01640                           *
01641                           *       1981 08 26
01642                           *
01643                           *    RELEASE LEVEL 6.75
01644                           *
01645                                   NAM    PCB7
01646                           *
01647                           * PCB7.AS  REV. 0.00
01648                           * STEVEN SHELTON
01649                           *
01650                           * THIS ROUTINE PROVIDES HOURLY DATA FOR THE O/P BUFFER
01651                           *
01652                           *
01652A C6A0                             ORG    PCB7
01654                           *
01655A C6A0 B6 0A     A         LDAB   HR        GET CURRENT TIME
01656A C6A2 27 02 C6A6          BEQ    070
01657A C6A4 20 02 C6A8          BRA    071
01658A C6A6 C6 24     A 070     LDAB   #024
01659A C6A8 C5 20     A 071     BITB   #020      START CORRECTION BCD-TO-BINARY
01660A C6AA 27 04 C6B0          BEQ    01
01661A C6AC C0 0C     A         SUBB   #12
01662A C6AE 20 04 C6B6          BRA    05
01663                           *
01664                           *        MOTOROLA M6800 CROSS-ASSEMBLER PAGE 2
01665                           *
01666A C6B0 C5 10     A 01      BITB   #010
01667A C6B2 27 02 C6B6          BEQ    05
01668A C6B4 C0 06     A         SUBB   #6
01669                           *
01670A C6B6 BE 9B     A 05      LDX    PTR       START OF HOUR VOL. BUFFERS
01671A C6B8 DF F8     A         STX    TEMP04
01672                           *
01673A C6BA 5A                  DECB
01674A C6BB 5B                  TSTB              COMPUTE ADDRESS OF HOUR BUFFER TO USE
01675A C6BC 27 0F C6CD          BEQ    09
01676A C6BE 96 F5     A         LDAA   SIZE
01677A C6C0 0C                  CLC
01678A C6C1 9B F9     A         ADDA   TEMP05
01679A C6C3 97 F9     A         STAA   TEMP05
01680A C6C5 24 03 C6CA          BCC    051
01681A C6C7 7C 00F8   A         INC    TEMP04
01682A C6CA 5A          051     DECB
```



```
01683A C6CB 20 EE C6BB          BRA    07
01684                           *
01685A C6CD DE F8     A 09      LDX    TEMP04
01686A C6CF B6 95     A         LDAB   SIZE
01687A C6D1 6F 00     A 011     CLR    0,X       CLEAR CURRENT BUFFER
01688A C6D3 08                  INX
01689A C6D4 5A                  DECB
01690A C6D5 26 FA C6D1          BNE    011
01691                           *
01692A C6D7 96 14     A         LDAA   NVOL1
01693A C6D9 97 F5     A         STAA   TEMP01
01694A C6DB 4F                  CLRA
01695A C6DC 97 F7     A         STAA   TEMP03
01696A C6DE 97 FA     A         STAA   TEMP06
01697A C6E0 97 FB     A         STAA   TEMP07
01698                           *
01699A C6E2 DE FA     A 050     LDX    TEMP06
01700                           *
01701A C6E4 A6 16     A         LDAA   VOLDW1,X
01702                           *
01703A C6E6 97 F6     A         STAA   TEMP02
01704A C6E8 86 01     A         LDAA   #001
01705A C6EA 97 A0     A         STAA   SPC1
01706A C6EC B6 F5     A         LDAB   TEMP01
01707A C6EE 27 06 C6F6          BEQ    015
01708A C6F0 86 70     A         LDAA   #070
01709A C6F2 97 A1     A         STAA   SPC2
01710A C6F4 20 08 C6FE          BRA    010
01711A C6F6 B6 15     A 015     LDAB   NVOL2
01712A C6F8 27 5A C754          BEQ    060
01713A C6FA 86 F0     A         LDAA   #0F0
01714A C6FC 97 A1     A         STAA   SPC2
01715                           *
01716A C6FE 96 F6     A 010     LDAA   TEMP02
01717A C700 DE A0     A         LDX    SPC1
01718A C702 96 F6     A 020     LDAA   TEMP02
01719A C704 0C                  CLC
01720A C705 48                  ASLA
01721A C706 97 F6     A         STAA   TEMP02
01722A C708 25 04 C70E          BCS    045
01723A C70A 27 34 C740          BEQ    030
01724A C70C 24 28 C736          BCC    040
01725                           *
01726A C70E A6 08     A 045     LDAA   VD1H,X
01727A C710 E6 09     A         LDAB   VD1L,X
01728A C712 6F 08     A         CLR    VD1H,X
01729A C714 6F 09     A         CLR    VD1L,X
01730A C716 36                  PSHA
01731A C717 96 F7     A         LDAA   TEMP03
01732A C719 7F 00F7   A         CLR    TEMP03
01733A C71C 0C                  CLC
01734A C71D 9B F9     A         ADDA   TEMP05
01735A C71F 97 F9     A         STAA   TEMP05
01736A C721 24 03 C726          BCC    052
01737A C723 7C 00F8   A         INC    TEMP04
01738A C726 32          052     PULA
01739A C727 DE F8     A         LDX    TEMP04
01740                           *
```

```
PAGE  031  YOUR    .SA10  PCS7

01741A C729 AB 00     A        ABDA   0,X
01742A C72B A7 00     A        STAA   0,X
01743                    *
01744A C72D 0C                 CLC
01745A C72E E9 01     A        ABCB   1,X
01746A C730 E7 01     A        STAB   1,X
01747A C732 24 02 C736         BCC    840
01748A C734 6C 00     A        INC    0,X
01749                    *
01750A C736 96 A1     A 840    LBAA   $PC2
01751A C738 80 10     A        SUBA   #$10
01752A C73A 97 A1     A        STAA   $PC2
01753A C73C BE A0     A        LBX    $PC1
01754A C73E 20 C2 C702         BRA    820
01755                    *
01756A C740 C6 02     A 830    LBAB   #02
01757A C742 B7 F7     A        STAB   TEMP03
01758A C744 B6 F5     A 835    LBAB   TEMP01
01759A C746 27 03 C74B         BEQ    836
01760A C748 5A                 DECB
01761A C749 B7 F5     A        STAB   TEMP01
01762A C74B 96 FB     A 836    LBAA   TEMP07
01763A C74D 4C                 INCA
01764A C74E 97 FB     A        STAA   TEMP07
01765A C750 80 09     A        SUBA   #$09
01766A C752 26 8E C6E2         BNE    850
01767                    *
01768A C754 39           840    RTS
01769                    *
01770A C755    FF     A        FCB    $0FF
01771                    *
01772                    *         MOTOROLA M68SAM CROSS-ASSEMBLER PAGE 1
01773                    *
01774                    * M68SAM IS THE PROPERTY OF MOTOROLA SPB, INC.
01775                    *        COPYRIGHT 1974 BY MOTOROLA INC
01776                    *
01777                    * MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
01778                    * CALTRANS TO M.T.C MODIFIED, RELEASE 2.0
01779                    *
01780                    * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
01781                    *
01782                    *           1981 08 08
01783                    *
01784                    *     RELEASE LEVEL 6.75
01785                    *
01786                          NAM    PCS1A
01787                    * PCS1A. AS   REV. 1.00
01788                    * STEVEN SKELTON
01789                    *
01790                    * FOR TYPE 334 CABINET
01791                    *
01792                    * THIS ROUTINE PROVIDES CORRESPONDENCE BETWEEN HARDWARE DETECTOR
01793                    * ASSIGNMENTS (PORT NO. & BIT NO.) AND THE INTERNAL SOFTWARE
01794                    * DETECTOR ASSIGNMENTS
01795                    *
01796A CA00                    ORG    INPUTT
01797                    *
01798A CA00 86 50     A        LBAA   #$50
```

```
PAGE  032  LUCK    .SA10  PCS1A

01799A CA02 97 F5     A          STAA   TEMP01   SET UP MSB OF PORT ADDRESSES
01800A CA04 4F                   CLRA            SET POINTER FOR DETECTOR INPUT PROCESSING AREAS
01801A CA05 97 EA     A          STAA   INDEX
01802A CA07 97 EB     A          STAA   INDEX+1
01803                    *
01804                    *        MOTOROLA M68SAM CROSS-ASSMBLER PAGE 2
01805                    *
01806A CA09 97 CC     A          STAA   DATAOM+020 CLEAR DETECTOR INPUT AREA
01807A CA0B 97 BC     A          STAA   DATAOM+030
01808A CA0D CE CA3F   A          LDX    #DTRTB   POINTER TO DETECTOR HARDWARE ASSIGNMENT TABLE
01809A CA10 BF F8     A          STX    TEMP04
01810A CA12 C6 01     A IP1      LDAB   #1       SOFTWARE DETECTOR POSITION MASK
01811                    *
01812A CA14 37           IP2      PSHB            SAVE SOFTWARE DETECTOR ID
01813A CA15 A6 00     A          LDAA   0,X      GET LSB OF PORT ADDRESS
01814A CA17 97 F6     A          STAA   TEMP02   PUT IT NEXT TO MSB
01815A CA19 A6 01     A          LDAA   1,X      GET THE HARDWARE DETECTOR POSITION MASK
01816A CA1B BE F5     A          LDX    TEMP01   INPUT PORT ADDRESS IN X
01817A CA1D A4 00     A          ANDA   0,X      SEE IF A CAR OVER DETECTOR
01818A CA1F 27 08 CA29           BEQ    NXDET    NO --
01819A CA21 BE EA     A          LDX    INDEX    YES -- GET SOFTWARE DETECTOR PROCESSING INDEX
01820A CA23 EA CC     A          ORAB   DATAOM+020,X GET THE INTERNAL DETECTOR BIT
01821A CA25 E4 E0     A          ANDB   DTCTRM+020,X KEEP ONLY ENABLED DETECTORS
01822A CA27 E7 CC     A          STAB   DATAOM+020,X PUT AWAY THE INTERNAL DETECTOR INPUT WORD
01823                    *
01824A CA29 BE F8     A NXDET    LDX    TEMP04   HARDWARE DETECTMP TABLE INDEX
01825A CA2B 08                   INX
01826A CA2C 08                   INX             FIX THE INDEX FOR NEXT ENTRY
01827A CA2D BF F8     A          STX    TEMP04   SAVE THE INDEX
01828A CA2F 33                   PULB            GET THE SOFTWARE DETECTOR POSITON BIT
01829                    *
01830A CA30 58                   ASLB            SHIFT IT LEFT
01831A CA31 24 E1 CA14           BCC    IP2      NOT DONE WITH THIS GROUP YET
01832A CA33 7D 00EB   A IP5      TST    INDEX+1  ARE WE ON FIRST GROUP OF DETECTORS?
01833A CA36 26 06 CA3E           BNE    DO       NO -- WE'RE DONE
01834A CA38 86 10     A IP3      LDAA   #$10     YES -- LETS FIX SOFTWARE DETECTOR AREA INDEX FOR SECOND GROUP
01835A CA3A 97 EB     A          STAA   INDEX+1
01836A CA3C 20 B4 CA12           BRA    IP1      GO DO THIS GROUP
01837                    *
01838A CA3E 39           DO       RTS             GO PROCESS THE DETECTORS
01839                    *
01840                    *
01841                    *        MMTOROLA M68SAM CROSS-ASSMBLER PAGE 3
01842                    *
01843A CA3F    02     A DTRTB    FCB    #02      DIRECTION ONE
01844A CA40    01     A          FCB    TBIT0    LANE NINE
01845A CA41    02     A          FCB    #02
01846A CA42    02     A          FCB    TBIT1    LANE TEN
01847A CA43    07     A          FCB    #02
01848A CA44    04     A          FCB    TBIT2    LANE ELEVEN
01849A CA45    02     A          FCB    #02
01850A CA46    08     A          FCB    TBIT3    LANE TWELEVE
01851A CA47    02     A          FCB    #02
01852A CA48    10     A          FCB    TBIT4    LANE THIRTEEN
01853A CA49    02     A          FCB    #02
01854A CA4A    20     A          FCB    TBIT5    LANE FOURTEEN
01855A CA4B    02     A          FCB    #02
01856A CA4C    40     A          FCB    TBIT6    LANE FIFTEEN
```

```
PAGE  033  LUCK    .SAIO  PCS1A

01857A CA4B    02    A        FCB    002
01858A CA4E    80    A        FCB    TBIT7    LANE SIXTEEN
01859                  *
01860A CA4F    06    A        FCB    006      DIRECTION TWO
01861A CA50    01    A        FCB    TBIT0    LALE NINE
01862A CA51    06    A        FCB    006
01863A CA52    02    A        FCB    TBIT1    LANE TEN
01864A CA53    06    A        FCB    006
01865A CA54    04    A        FCB    TBIT2    LANE ELEVEN
01866A CA55    06    A        FCB    006
01867A CA56    08    A        FCB    TBIT3    LANE TWELEVE
01868A CA57    06    A        FCB    006
01869A CA58    10    A        FCB    TBIT4    LANE THIRTEEN
01870A CA59    06    A        FCB    006
01871A CA5A    20    A        FCB    TBIT5    LANE FOURTEEN
01872A CA5B    06    A        FCB    006
01873A CA5C    40    A        FCB    TBIT6    LANE FIFTEEN
01874A CA5D    06    A        FCB    006
01875A CA5E    80    A        FCB    TBIT7    LANE SIXTEEN
01876                  *
01877A CA5F    FFFF  A        FDB    00FFFF
01878A CA61    FFFF  A        FDB    00FFFF
01879A CA63    FF    A        FCB    00FF
01880                  *
01881                  *        MOTOROLA M6800 CROSS-ASSEMBLER PAGE 1
01882                  *
01883                  * M6800 IS THE PROPERTY OF MOTOROLA SPD, INC.
01884                  *        COPYRIGHT 1974 BY MOTOROLA INC
01885                  *
01886                  * MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
01887                  * CALTRANS TO M.T.C. MODIFIED, RELEASE 2.0
01888                  *
01889                  * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
01890                  *
01891                  *            1981 07 03
01892                  *
01893                  *      RELEASE LEVEL 6.75
01894                  *
01895                        NAM    PCS2
01896                  *
01897                  * PCS2 AS   REV. 1.00
01898                  * STEVEN SKELTON
01899                  *
01900                  * THIS ROUTINE PROCESSES THE DETECTOR INPUTS
01901                  *
01902                  * THE RAW INPUTS ARE VERIFIED AND THEN UTILIZED
01903                  * BY THE CONTROLLER.
01904                  *
01905                  * DATAOH AND DATAOL ARE THE CURRENT DETECTOR INPUTS
01906                  * AND DETINH AND DETINL ARE THE VERIFIED INPUTS.
01907                  *
01908                  *
01909                  *        MOTOROLA M6800 CROSS-ASSEMBLER PAGE 2
01910                  *
01911A C350                     ORG    PCS2
01912A C350 BD C2A0  A          JSR    INPUT    INPUT THE DETECTORS
01913A C353 BD CA00  A          JSR    INPUTT
01914A C356 20 06 C35E          BRA    S0
```

```
PAGE  034  LUCK    .SA10  PC82

01915A C358 01                  NOP
01916A C359 01                  NOP
01917A C35A 01                  NOP
01918A C35B 01                  NOP
01919A C35C 01                  NOP
01920                    *
01921                    * SINGLE SAMPLE ERROR CORRECTION ROUTINE
01922                    *
01923A C35D 01                  NOP
01924A C35E 86 02     A S0      LDAA   #2        TWICE THRU LOOP
01925A C360 97 F8     A         STAA   TEMP04    INITIALIZE LOOP COUNTER
01926A C362 4C                  INCA
01927A C363 97 F9     A         STAA   TEMP05
01928A C365 CE 00BC   A         LDX    #DATA0L
01929A C368 DF FA     A         STX    TEMP06
01930A C36A CE 00AC   A         LDX    #DATA0M   INITIALIZE INDEX REGISTER TO MSB'S
01931                    *
01932A C36D E6 00     A S1      LDAB   0,X       DATA0
01933A C36F E4 14     A         ANDB   BYCTRN-DATA0M,X MASK UNWANTED DETECTORS
01934A C371 37                  PSHB             SAVE DATA0
01935A C372 E8 01     A         EORB   1,X       DATA0. XOR. DATA1=(ACCB)
01936A C374 A6 01     A         LDAA   1,X       DATA1
01937A C376 A8 02     A         EORA   2,X       DATA1. XOR. DETIN(OLD)
01938A C378 A7 00     A         STAA   0,X       SAVE FOR NEXT OPERATION
01939A C37A E4 00     A         ANDB   0,X       (DATA0. XOR. DATA1).AND.(DATA1. XOR.DETIN)=(ACCB)
01940A C37C E8 01     A         EORB   1,X       (ACCB).XOR.DATA1 - NEW DETIN!
01941                    *
01942A C37E 32                  PULA             DATA0
01943A C37F A7 01     A         STAA   1,X       PUT IN DATA1 SLOT FOR NEXT TIME THROUGH
01944A C381 A8 02     A         EORA   2,X       DATA0. XOR. DETIN=(ACCA)
01945A C383 A4 00     A         ANDA   0,X       (DATA0. XOR. DETIN).AND.(DATA1.XOR.DETIN)=(ACCA)
01946A C385 A7 03     A         STAA   3,X       NEW STATE CHANGE WORD
01947A C387 E7 02     A         STAB   2,X       VERIFIED DETECTOR INPUT - DETIL
01948                    *
01949                    *        MOTOROLA M6800AM CROSS-ASSEMBLER PAGE 3
01950                    *
01951A C389 DE FA     A         LDX    TEMP06
01952A C38B 7A 00F8   A         DEC    TEMP04    DECREMENT LOOP COUNTER
01953A C38E 26 BD C36D          BNE    S1        FIRST TIME THROUGH GO BACK TO S1
01954A C390 96 FB     A         LDAA   TEMP07
01955A C392 8B 10     A         ADDA   #$10
01956A C394 97 FB     A         STAA   TEMP07
01957A C396 DE FA     A         LDX    TEMP06
01958A C398 7C 00F8   A         INC    TEMP04
01959A C39B 7A 00F9   A         DEC    TEMP05
01960A C39E 26 CD C36D          BNE    S1
01961                    *
01962A C3A0 39                  RTS
01963                    *
01964                    *          MOTOROLA M6800AM CROSS-ASSEMBLER PAGE 1
01965                    *
01966                    * M6800AM IS THE PROPERTY OF MOTOROLA SPD, INC.
01967                    *        COPYRIGHT 1974 BY MOTOROLA INC
01968                    *
01969                    * MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
01970                    * CALTRANS TO M.T.C. MODIFIED, RELEASE 2.0
01971                    *
01972                    * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
```

```
PAGE  035  LUCK    .SA10  PCS2

                        *
                        *              1981 07 03
                        *
                        *        RELEASE LEVEL 6.75
                        *
                                 NAM    PCS3
                        *
                        * PCS3. AS  REV. 1.00
                        * STEVEN SKELTON
                        *
                        *  UPSTREAM DETECTOR  SERVICING ROUTINE
                        *
                        * THIS ROUTINE SERVICES EIGHT LANE DETECTORS IN ONE DIRECTION.
                        * THE ROUTINE ADDS THE INDIVIDUAL COUNTS TO THE SIX SECOND
                        * OCCUPANCY AND VOLUME COUNTERS.
                        *
D4A0                            ORG    PCS3
D4A0 BD C060   A                JSR    SETUP
D4A3 86 80     A                LDAA   0TBIT7   LOAD THE LANE DETECTOR MASK
D4A5 97 F7     A                STAA   TEMP03
                        *
D4A7 BF F5     A T1             STX    TEMP01
D4A9 7C 00F5   A                INC    TEMP01
D4AC BE F5     A                LDX    TEMP01
D4AE 96 F7     A                LDAA   TEMP03
D4B0 95 B0     A                BITA   00
D4B2 27 0B D4BF                 BEQ    T20
D4B4 0C                         CLC
D4B5 86 01     A                LDAA   001
D4B7 A9 00     A                ADCA   B1E,X
D4B9 A7 00     A                STAA   B1E,X
D4BB 24 02 D4BF                 BCC    T20
D4BD 6C 08     A                INC    C1E,X
D4BF 96 F7     A T20            LDAA   TEMP03
D4C1 95 B1     A                BITA   OUT
D4C3 27 02 D4C7                 BEQ    T21
D4C5 6C 01     A                INC    D1B,X
                        *
D4C7 BF F5     A T21            STX    TEMP01
D4C9 7A 00F5   A                DEC    TEMP01
D4CC BE F5     A                LDX    TEMP01
D4CE 95 AE     A                BITA   DETINM   IS THIS M.L. DETECTOR OCCUPIED?
D4D0 26 1A D4EC                 BNE    T16      IF NO, GO TO L16
                        *
                        *        MOTOROLA M68BAM CROSS-ASSEMBLER PAGE 2
                        *
D4D2 95 AF     A T2             BITA   STCHGM   HAS THE STATE CHANGED?
D4D4 27 22 D4F8                 BEQ    T14      IF NOT, GET ANOTHER LANE
                        *
D4D6 6C 01     A                INC    VSIX1,X  INCREMENT THIS LANE'S SIX SECOND VOLUME
                        *
D4D8 16                         TAB
D4D9 53                         COMB
D4DA B4 AF     A                ANDB   STCHGM
D4DC B7 AF     A                STAB   STCHGM
                        *
D4DE 9A B0     A                ORAA   00
D4E0 97 B0     A                STAA   00
```

```
02031                          *
02032A D4E2 6F 08     A        CLR    STCHB1,X  CLEAR STATE CHANGE COUNT
02032A D4E4 C6 BF     A        LDAB   #$0BF     MASK FOR ACTIVE STATE CHANGE ERROR BIT
02034A D4E6 E4 0A     A        ANDB   BET1,X    RESET THE BIT
02035A D4E8 E7 0A     A        STAB   BET1,X    SAVE THE OTHERS
02036A D4EA 20 0C D4F8         BRA    T14       CHECK ANOTHER LANE
02037                          *
02038A D4EC 6C 00     A T16    INC    OBIX1,X   INCREMENT OCCUPANCY COUNT
02039                          *
02040A D4EE 6C 0F     A        INC    OCC,X
02041A D4F0 95 AF     A        BITA   STCHGM
02042A D4F2 27 04 D4F8         BEQ    T14
02043                          *
02044A D4F4 9A B1     A        ORAA   OOT
02045A D4F6 97 B1     A        STAA   OOT
02046                          *
02047A D4F8 44                 T14    LSRA             SHIFT DETECTOR MASK FOR NEXT LANE
02048A D4F9 97 F7     A        STAA   TEMP03
02049A D4FB BD C030   A        JSR    NEXTLN    ARE WE DONE?
02050A D4FE 24 A7 D4A7         BCC    T1        IF NOT FINISHED, GO BACK FOR NEXT LANE
02051A D000 39                 RTS
02052                          *
02053                          *          MOTOROLA M6800AN CROSS-ASSEMBLER PAGE 1
02054                          *
02055                          * M6800AN IS THE PROPERTY OF MOTOROLA SPD, INC.
02056                          *        COPYRIGHT 1974 BY MOTOROLA INC
02057                          *
02058                          * MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
02059                          * CALTRANS TO M.T.C. MODIFIED, RELEASE 2.0
02060                          *
02061                          * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
02062                          *
02063                          *             1981 07 03
02064                          *
02065                          *       RELEASE LEVEL 6.75
02066                          *
02067                                 MAN    PCS3A
02068                          *
02069                          * PCS3A. AS  REV. 1.00
02070                          * STEVEN SKELTON
02071                          *
02072                          *  UPSTREAM DETECTOR  SERVICING ROUTINE
02073                          *
02074                          * THIS ROUTINE SERVICES EIGHT LANE DETECTORS IN ONE DIRECTION.
02075                          * THE RMUTINE ADDS THE INDIVIDUAL COUNTS TO THE SIX SECOND
02076                          * OCCUPANCY AND VOLUME COUNTERS.
02077                          *
02078A 0550                           ORG    PCS3A
02079                          *
02080A 0550 BD C060   A        JSR    SETUP     POSITION X REG. FOR MAIN LANE DATA
02081A 0553 86 80     A        LDAA   #TBIT7    LOAD THE LANE DETECTOR MASK
02082A 0555 97 F7     A        STAA   TEMP03
02083                          *
02084A 0557 BF F5     A TA1    STX    TEMP01
02085A 0559 7C 00F5   A        INC    TEMP01
02086A 055C BE F5     A        LDX    TEMP01
02087A 055E 96 F7     A        LDAA   TEMP03
02088A 0560 95 82     A        BITA   #04
```

```
PAGE  027  LUCK    .8A0D  PC83A

020094A 8562 27 08 8568          BEQ    TA20
020090A 8564 0C                  CLC
020091A 8565 B6 01     A         LDAA   001
020092A 8567 A9 02     A         ADCA   B2E,X
020093A 8569 A7 02     A         STAA   B2E,X
020094A 856B 24 02 8568          BCC    TA20
020095A 856D 6C 06     A         INC    C2E,X
020096A 856F 96 F7     A TA20    LDAA   TEMP03
020097A 8571 95 B3     A         BITA   00TA
020098A 8573 27 02 8577          BEQ    TA21
020099A 8575 6C 03     A         INC    B3B,X
02100                     0
02101A 8577 DF F5      A TA21    STX    TEMP01
02102A 8579 7A 00F5    A         DEC    TEMP01
02103A 857C DE F5      A         LDX    TEMP01
02104A 857E 95 BE      A         BITA   BETIML    IS THIS M.L. DETECTOR OCCUPIED?
02105A 8580 26 14 8596           BNE    TA14      IF SO, GO TO L14
02106                     0
02107                     0          MOTOROLA M6800 CROSS-ASSEMBLER PAGE 2
02108                     0
02109A 8582 95 BF      A TA2     BITA   STCHBL    HAS THE STATE CHANGED?
02110A 8584 27 1C 85A2           BEQ    TA14      IF NOT, GET ANOTHER LANE
02111                     0
02112A 8586 6C 81      A         INC    VBIX1+080,X INCREMENT THIS LANE'S SIX SECOND VOLUME
02113                     0
02114A 8588 96 B2      A         ORAA   00A
02115A 858A 97 B2      A         STAA   00A
02116                     0
02117A 858C 6F 88      A         CLR    STCHB1+080,X CLEAR STATE CHANGE COUNT
02118A 858E C6 BF      A         LDAB   00080F     MASK FOR ACTIVE STATE CHANGE ERROR BIT
02119A 8590 E4 8A      A         ANDB   DET1+080,X RESET THE BIT
02120A 8592 E7 8A      A         STAB   DET1+080,X SAVE THE OTHERS
02121A 8594 20 0C 85A2           BRA    TA14      CHECK ANOTHER LANE
02122                     0
02122A 8596 6C 86      A TA14    INC    OBIX1+080,X INCREMENT OCCUPANCY COUNT
02124                     0
02125A 8598 6C 8F      A         INC    OCC+080,X
02126A 859A 95 BF      A         BITA   STCHBL
02127A 859C 27 04 85A2           BEQ    TA14
02128A 859E 9A B3      A         ORAA   00TA
02129A 85A0 97 B3      A         STAA   00TA
02130                     0
02131A 85A2 44            TA14   LSRA             SHIFT DETECTOR MASK FOR NEXT LANE
02132A 85A3 97 F7      A         STAA   TEMP03
02133A 85A5 BD C020    A         JSR    NEXTLN    ARE LE DONE?
02134A 85A8 24 AB 8557           BCC    TA1       IF NOT FINISHED, GO BACK FOR NEXT LANE
02135A 70AA 39                   RTS
02136                     0
02137                     0          MOTOROLA M6800 CROSS-ASSEMBLER PAGE 1
02138                     0
02139                     0 M6800 IS THE PROPERTY OF MOTOROLA SPD, INC.
02140                     0        COPYRIGHT 1974 BY MOTOROLA INC
02141                     0
02142                     0 MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.0
02143                     0 CALTRANS TO N.T.C. MODIFIED, RELEASE 2.0
02144                     0
02145                     0 THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
02146                     0
```

```
PAGE  028  LUCK    .BA10  PCS3A

                    *              1982 03 16
                    *
                    *        RELEASE LEVEL 6.75
                    *
                           NAM    PCS3B
                    *
                    * PCS3B. AS  REV. 0.00
                    * STEVEN SKELTON
                    *
                    *  DOWNSTREAM DETECTOR  SERVICING ROUTINE
                    *
                    * THIS ROUTINE SERVICES EIGHT LANE DETECTORS IN ONE DIRECTION.
                    *
CBB0                       ORG    PCS3B
CBB0 BD C060  A            JSR    SETUP
CBB3 86 80    A            LDAA   #TBIT7   LOAD THE LANE DETECTOR MASK
                    *
CBB5 95 CE    A TB1        BITA   SE:INM+#20 IS THIS M.L. DETECTOR OCCUPIED?
CBB7 26 08 CBC6            BNE    TB16     IF SO, GO TO L16
                    *
                    *        MOTOROLA M6800 CROSS-ASSEMBLER PAGE 2
                    *
CBB9 95 CF    A TB2        BITA   STCHGR+#20 HAS THE STATE CHANGED?
CBBB 27 14 CBD1            BEQ    TB14     IF NOT, GET ANOTHER LANE
                    *
CBBD 16                    TAB
CBBE 43                    COMA
CBBF 94 80    A            ANDA   GO
CBC1 97 80    A            STAA   GO
CBC3 17                    TBA
CBC4 20 08 CBD1            BRA    TB14
                    *
CBC6 95 CF    A TB16       BITA   STCHGR+#20
CBC8 27 07 CBD1            BEQ    TB14
CBCA 16                    TAB
CBCB 43                    COMA
CBCC 94 81    A            ANDA   GOT
CBCE 97 81    A            STAA   GOT
CBD0 17                    TBA
                    *
CBD1 44            TB14    LSRA           SHIFT DETECTOR MASK FOR NEXT LANE
CBD2 80 C030  A            JSR    NEXTLN  ARE WE DONE?
CBD5 24 BE CBB5            BCC    TB1     IF NOT FINISHED, GO BACK FOR NEXT LANE
CBD7 39                    RTS
                    *
                    *          MOTOROLA M6800 CROSS-ASSEMBLER PAGE 1
                    *
                    * M6800 IS THE PROPERTY OF MOTOROLA SPD, INC.
                    *        COPYRIGHT 1974 BY MOTOROLA INC
                    *
                    * MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
                    * CALTRANS TO M.T.C. MODIFIED, RELEASE 2.0
                    *
                    * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
                    *
                    *              1982 03 16
                    *
                    *        RELEASE LEVEL 6.75
```

```
PAGE  039  LUCK    .SA10  PCS3B

                        *
                              NAM    PCS3C
                        *
                        * PCS3C. 4B  REV. 0.00
                        * STEVEN SKELTON
                        *
                        *  DOWNSTREAM DETECTOR  SERVICING ROUTINE
                        *
                        * THIS ROUTINE SERVICES EIGHT LANE DETECTORS IN ONE DIRECTION.
                        *
CC00                          ORG    PCS3C
                        *
CC00 BD C060  A               JSR    SETUP    POSITION X REG. FOR MAIN LANE DATA
CC03 86 80    A               LDAA   #TBIT7   LOAD THE LANE DETECTOR MASK
                        *
CC05 95 BE    A TC1           BITA   DETINL+#20 IS THIS M.L. DETECTOR OCCUPIED?
CC07 26 0B CC16               BNE    TC16     IF SO, GO TO L16
                        *
                        *        MOTOROLA M6800AM CROSS-ASSEMBLER PAGE 2
                        *
CC09 95 BF    A TC2           BITA   STCHGL+#20 HAS THE STATE CHANGED?
CC0B 27 14 CC21               BEQ    TC14     IF NOT, GET ANOTHER LANE
                        *
CC0D 16                       TAB
CC0E 43                       COMA
CC0F 94 B3    A               ANDA   DDA
CC11 97 B3    A               STAA   DDA
CC13 17                       TBA
CC14 20 0B CC21               BRA    TC14     CHECK ANOTHER LANE
                        *
CC16 95 BF    A TC16          BITA   STCHGL+#20
CC18 27 07 CC21               BEQ    TC14
CC1A 16                       TAB
CC1B 43                       COMA
CC1C 94 B3    A               ANDA   DDTA
CC1E 97 B3    A               STAA   DDTA
CC20 17                       TBA
                        *
CC21 44                 TC14  LSRA            SHIFT DETECTOR MASK FOR NEXT LANE
CC22 BD C030  A               JSR    NEXTLN   ARE WE DONE?
CC25 24 DE CC05               BCC    TC1      IF NOT FINISHED, GO BACK FOR NEXT LANE
CC27 39                       RTS
                        *
                        *        MOTOROLA M6800AM CROSS-ASSEMBLER PAGE 1
                        *
                        * M6800AM IS THE PROPERTY OF MOTOROLA SPD, INC.
                        * COPYRIGHT 1974 BY MOTOROLA INC
                        *
                        * MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
                        * WRITTEN BY M.T.C. , RELEASE 2.0
                        *
                        * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
                        *
                        *       1982 03 16
                        *
                        *    RELEASE LEVEL 6.75
                        *
                              NAM    PCS7A
```

```
02263                    *
02264                    * PCS7A AS  REV. 0.00
02265                    * STEVEN SKELTON
02266                    *
02267                    * THIS ROUTINE PROVIDES HOURLY DATA FOR THE O/P BUFFER
02268                    *
02269                    *
02270A CF00                        ORG    PCS7A
02271                    *
02272A CF00 B6 0A     A            LDAB   HR        GET CURRENT TIME
02273A CF02 27 02 CF06             BEQ    GA70
02274A CF04 20 02 CF08             BRA    GA71
02275A CF06 C6 24     A GA70       LDAB   #$24
02276A CF08 C5 20     A GA71       BITB   #$20      START CORRECTION BCD-TO-BINARY
02277A CF0A 27 04 CF10             BEQ    GA1
02278A CF0C C0 0C     A            SUBB   #12
02279A CF0E 20 06 CF16             BRA    GA5
02280                    *
02281                    *         MOTOROLA M68SAM CROSS-ASSMBLER PAGE 2
02282                    *
02283A CF10 C5 10     A GA1        BITB   #$10
02284A CF12 27 02 CF16             BEQ    GA5
02285A CF14 C0 06     A            SUBB   #6
02286                    *
02287A CF16 DE 96     A GA5        LDX    PTRL      START OF LENGTH DATA  BUFFERS
02288A CF18 DF F8     A            STX    TEMP04
02289                    *
02290A CF1A 5A                     DECB
02291A CF1B 5D          GA7        TSTB              COMPUTE ADDRESS OF HOUR BUFFER TO USE
02292A CF1C 27 0F CF2D             BEQ    GA9
02293A CF1E 96 95     A            LDAA   SIZE
02294A CF20 0C                     CLC
02295A CF21 9B F9     A            ADDA   TEMP05
02296A CF23 97 F9     A            STAA   TEMP05
02297A CF25 24 03 CF2A             BCC    GA51
02298A CF27 7C 00F8   A            INC    TEMP04
02299A CF2A 5A          GA51       DECB
02300A CF2B 20 EE CF1B             BRA    GA7
02301                    *
02302A CF2D DE F8     A GA9        LDX    TEMP04
02303A CF2F B6 95     A            LDAB   SIZE
02304A CF31 6F 00     A GA11       CLR    0,X       CLEAR CURRENT BUFFER
02305A CF33 08                     INX
02306A CF34 5A                     DECB
02307A CF35 26 FA CF31             BNE    GA11
02308                    *
02309A CF37 96 14     A            LDAA   NVOL1
02310A CF39 97 F5     A            STAA   TEMP01
02311A CF3B 4F                     CLRA
02312A CF3C 97 F7     A            STAA   TEMP03
02313A CF3E 97 F6     A            STAA   TEMP06
02314A CF40 97 FB     A            STAA   TEMP07
02315                    *
02316A CF42 DE FA     A GA50       LDX    TEMP06
02317                    *
02318A CF44 A6 16     A            LDAA   VOLDW1,X
02319                    *
02320A CF46 97 F6     A            STAA   TEMP02
```



```
02321A CF48 86 01     A            LDAA   #$01
02322A CF4A 97 A0     A            STAA   SPC1
02323A CF4C B6 F5     A            LDAB   TEMP01
02324A CF4E 27 06 CF56             BEQ    GA15
02325A CF50 86 70     A            LDAA   #$70
02326A CF52 97 A1     A            STAA   SPC2
02327A CF54 20 08 CF5E             BRA    GA10
02328A CF56 B6 15     A GA15       LDAB   NVOL2
02329A CF58 27 5A CFB4             BEQ    GA60
02330A CF5A 86 F0     A            LDAA   #$F0
02331A CF5C 97 A1     A            STAA   SPC2
02332                    *
02333A CF5E 96 F6     A GA10       LDAA   TEMP02
02334A CF60 DE A0     A            LDX    SPC1
02335A CF62 96 F6     A GA20       LDAA   TEMP02
02336A CF64 0C                     CLC
02337A CF65 48                     ASLA
02338A CF66 97 F6     A            STAA   TEMP02
02339A CF68 25 04 CF6E             BCS    GA45
02340A CF6A 27 34 CFA0             BEQ    GA30
02341A CF6C 24 28 CF96             BCC    GA40
02342                    *
02343A CF6E A6 0D     A GA45       LDAA   LENH,X
02344A CF70 E6 0E     A            LDAB   LENL,X
02345A CF72 6F 0D     A            CLR    LENH,X
02346A CF74 6F 0E     A            CLR    LENL,X
02347A CF76 36                     PSHA
02348A CF77 96 F7     A            LDAA   TEMP03
02349A CF79 7F 00F7   A            CLR    TEMP03
02350A CF7C 0C                     CLC
02351A CF7D 9B F9     A            ADDA   TEMP05
02352A CF7F 97 F9     A            STAA   TEMP05
02353                    *
02354A CF81 24 03 CF86             BCC    GA52
02355A CF83 7C 00F8   A            INC    TEMP04
02356A CF86 32          GA52       PULA
02357A CF87 DE F8     A            LDX    TEMP04
02358                    *
02359A CF89 AB 00     A            ADDA   0,X
02360A CF8B A7 00     A            STAA   0,X
02361                    *
02362A CF8D 0C                     CLC
02363A CF8E E9 01     A            ADCB   1,X
02364A CF90 E7 01     A            STAB   1,X
02365A CF92 24 02 CF96             BCC    GA40
02366A CF94 6C 00     A            INC    0,X
02367                    *
02368A CF96 96 A1     A GA40       LDAA   SPC2
02369A CF98 80 10     A            SUBA   #$10
02370A CF9A 97 A1     A            STAA   SPC2
02371A CF9C DE A0     A            LDX    SPC1
02372A CF9E 20 C2 CF62             BRA    GA20
02373                    *
02374A CFA0 C6 02     A GA30       LDAB   #02
02375A CFA2 D7 F7     A            STAB   TEMP03
02376A CFA4 B6 F5     A GA35       LDAB   TEMP01
02377A CFA6 27 03 CFAB             BEQ    GA36
02378A CFA8 5A                     DECB
```

```
02379A CFA9 B7 F5     A          STAB   TEMP01
02380A CFAB 96 FB     A GA36     LDAA   TEMP07
02381A CFAD 4C                   INCA
02382A CFAE 97 FB     A          STAA   TEMP07
02383A CFB0 80 09     A          SUBA   #$09
02384A CFB2 26 BE CF42           BNE    GA50
02385                   *
02386A CFB4 39          GA60     RTS
02387                   *
02388A CFB5    FF     A          FCB    $0FF
02389                   *
02390                   *          MOTOROLA M6800AN CROSS-ASSEMBLER PAGE 1
02391                   *
02392                   * M6800AN IS THE PROPERTY OF MOTOROLA SPD, INC.
02393                   * COPYRIGHT 1974 BY MOTOROLA INC
02394                   *
02395                   * MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
02396                   * WRITTEN BY M.T.C. , RELEASE 2.0
02397                   *
02398                   * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
02399                   *
02400                   *       1983 03 17
02401                   *
02402                   *    RELEASE LEVEL 6.75
02403                   *
02404                           NAM    PCS8
02405                   *
02406                   * PCS8 A8  REV. 0.00
02407                   * STEVEN SKELTON
02408                   *
02409                   * THIS ROUTINE ACCOUNTS FOR VEHICLE SPEED CLASSIFICATION.
02410                   * IT TREATS LANES AFFECTED BY PCS3 AND PCS3B.
02411                   *
02412A CC50                      ORG    PCS8
02413                   *
02414A CC50 86 07     A          LDAA   #7
02415A CC52 97 E8     A          STAA   COUNT6
02416A CC54 96 E5     A          LDAA   COUNT3
02417A CC56 27 04 CC5C           BEQ    SP99
02418A CC58 7F 00E5   A          CLR    COUNT3
02419A CC5B 39                   RTS
02420A CC5C 86 01     A SP99     LDAA   #01
02421A CC5E 97 E5     A          STAA   COUNT3
02422                   *
02423A CC60 CE 0270   A          LDX    #$0270
02424A CC63 86 80     A          LDAA   #$80
02425A CC65 DF EA     A          STX    INDEX
02426A CC67 97 E3     A          STAA   COUNT1
02427A CC69 96 E3     A SP17     LDAA   COUNT1
02428A CC6B 94 80     A          ANDA   #80
02429A CC6D 26 47 CCB6           BNE    SP11
02430A CC6F E6 08     A          LDAB   C1E,X
02431A CC71 27 0A CC7D           BEQ    SP49
02432A CC73 6F 08     A SP59     CLR    C1E,X
02433A CC75 6F 00     A          CLR    B1E,X
02434A CC77 6F 0F     A          CLR    OCC,X
02435A CC79 6F 01     A          CLR    B1B,X
02436A CC7B 20 59 CCB6           BRA    SP11
```



```
02437A CC7D E6 00     A SP49     LDAB   B1E,X
02438A CC7F 27 55 CCB6           BEQ    SP11
02439A CC81 D1 E8     A          CMPB   COUNT6
02440A CC83 2B EE CC73           BMI    SP59
02441A CC85 6F 00     A          CLR    B1E,X
02442A CC87 DF F6     A          STX    TEMP02
02443A CC89 D7 F8     A          STAB   TEMP04
02444A CC8B A6 04     A          LDAA   LB,X
02445A CC8D 97 F9     A          STAA   TEMP05
02446A CC8F BD C9E0   A          JSR    DIV
02447A CC92 D6 FA     A          LDAB   TEMP06
02448A CC94 D7 F5     A          STAB   TEMP01
02449A CC96 DE F6     A          LDX    TEMP02
02450A CC98 E6 01     A          LDAB   B1B,X
02451A CC9A 6F 01     A          CLR    B1B,X
02452A CC9C DF F6     A          STX    TEMP02
02453A CC9E D7 F8     A          STAB   TEMP04
02454A CCA0 A6 04     A          LDAA   LB,X
02455A CCA2 97 F9     A          STAA   TEMP05
02456A CCA4 BD C9E0   A          JSR    DIV
02457                   *
02458A CCA7 86 01     A          LDAA   #01
02459A CCA9 97 EA     A          STAA   INDEX
02460A CCAB DE EA     A          LDX    INDEX
02461                   *
02462                   *
02463A CCAD D6 FA     A          LDAB   TEMP06
02464A CCAF DB F5     A          ADDB   TEMP01
02465A CCB1 37                   ASPB
02466A CCB2 D7 F5     A          STAB   TEMP01
02467A CCB4 D7 F8     A          STAB   TEMP04
02468A CCB6 A6 0F     A          LDAA   OCC,X
02469A CCB8 6F 0F     A          CLR    OCC,X
02470A CCBA 97 F9     A          STAA   TEMP05
02471A CCBC BD C990   A          JSR    MULT
02472A CCBF D6 FB     A          LDAB   TEMP07
02473                   *
02474A CCC1 86 02     A          LDAA   #02
02475A CCC3 97 EA     A          STAA   INDEX
02476A CCC5 DE EA     A          LDX    INDEX
02477                   *
02478A CCC7 E0 06     A          SUBB   LB,X
02479A CCC9 D1 A3     A          CMPB   TRUCK
02480A CCCB 2B 17 CCE4           BMI    SP2
02481A CCCD 0C                   CLC
02482                   *
02483A CCCE 86 01     A          LDAA   #01
02484A CCD0 97 EA     A          STAA   INDEX
02485A CCD2 DE EA     A          LDX    INDEX
02486                   *
02487A CCD4 20 04 CCDA           BRA    SP9
02488A CCD6 20 56 CD2E SP11      BRA    SP1
02489A CCD8 20 8F CC69 SP7       BRA    SP17
02490A CCDA A6 0E     A SP9      LDAA   LENL,X
02491A CCDC 89 01     A          ADCA   #1
02492A CCDE A7 0E     A          STAA   LENL,X
02493A CCE0 24 02 CCE4           BCC    SP2
02494A CCE2 6C 0D     A          INC    LENH,X
```

```
02495                          *
02496A CCE4 CE 0200   A SP2    LDX    #00200
02497A CCE7 D6 F5     A        LDAB   TEMP01
02498A CCE9 E1 00     A SP4    CMPB   SPD,X
02499A CCEB 2E 39 CD16         BGT    SP12
02500A CCED 2A 20 CD0F         BPL    SP3
02501A CCEF 08            SP6  INX
02502A CCF0 8C 020B   A        CPX    #0020B
02503A CCF3 26 F4 CCE9         BNE    SP4
02504A CCF5 0C                 CLC
02505A CCF6 A6 C0     A        LDAA   SPED+020,X
02506A CCF8 89 01     A        ADCA   #1
02507A CCFA A7 C0     A        STAA   SPED+020,X
02508A CCFC 24 0F CD0D         BCC    SP5
02509A CCFE 0C                 CLC
02510A CCFF A6 B0     A        LDAA   SPED+010,X
02511A CD01 89 01     A        ADCA   #1
02512A CD03 A7 B0     A        STAA   SPED+010,X
02513A CD05 24 06 CD0D         BCC    SP5
02514A CD07 A6 A0     A        LDAA   SPED,X
02515A CD09 8B 01     A        ADDA   #1
02516A CD0B A7 A0     A        STAA   SPED,X
02517A CD0D 20 1F CD2E SP5     BRA    SP1
02518                          *
02519A CD0F 96 FA     A SP3    LDAA   TEMP03
02520A CD11 A1 90     A        CMPA   REM,X
02521A CD13 2A 01 CD16         BPL    SP12
02522A CD15 08                 INX
02523
02524A CD16 0C            SP12 CLC
02525A CD17 A6 C0     A        LDAA   SPED+020,X
02526A CD19 89 01     A        ADCA   #1
02527A CD1B A7 C0     A        STAA   SPED+020,X
02528A CD1D 24 0F CD2E         BCC    SP1
02529A CD1F 0C                 CLC
02530A CD20 A6 B0     A        LDAA   SPED+010,X
02531A CD22 89 01     A        ADCA   #1
02532A CD24 A7 B0     A        STAA   SPED+010,X
02533A CD26 24 06 CD2E         BCC    SP1
02534A CD28 A6 A0     A        LDAA   SPED,X
02535A CD2A 8B 01     A        ADDA   #1
02536A CD2C A7 A0     A        STAA   SPED,X
02537                          *
02538A CD2E 96 E3     A SP1    LDAA   COUNT1
02539A CD30 47                 ASRA
02540A CD31 97 E3     A        STAA   COUNT1
02541A CD33 86 02     A        LDAA   #02
02542A CD35 97 EA     A        STAA   INDEX
02543A CD37 BD C030   A        JSR    NEXTLN
02544A CD3A 24 9C CCD8         BCC    SP7
02545A CD3C 39                 RTS
02546                          *
02547                          *
02548                          *          MOTOROLA M68SAM CROSS-ASSEMBLER PAGE 1
02549                          *
02550                          * M68SAM IS THE PROPERTY OF MOTOROLA SPD, INC.
02551                          * COPYRIGHT 1974 BY MOTOROLA INC
02552                          *
```



```
02553                          * MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.1
02554                          * WRITTEN BY M.T.C. , RELEASE 2.0
02555                          *
02556                          * THIS PROGRAM MATERIAL FOR DESIGN STUDY ONLY
02557                          *
02558                          *       1982 03 17
02559                          *
02560                          *    RELEASE LEVEL 4.75
02561                          *
02562                                  MAM    PCSBA
02563                          *
02564                          * PCSBA AS  REV. 0.00
02565                          * STEVEN SKELTON
02566                          *
02567                          * THIS ROUTINE ACCOUNTS FOR VEHICLE SPEED CLASSIFICATION.
02568                          * IT TREATS LANES AFFECTED BY PCS3A AND PCS3C.
02569                          *
02570A CDA0                            ORG    PCSBA
02571                          *
02572A CDA0 86 07     A        LDAA   #7
02573A CDA2 97 E8     A        STAA   COUNT6
02574A CDA4 96 E6     A        LDAA   COUNT4
02575A CDA6 27 04 CDAC         BEQ    SPA99
02576A CDA8 7F 00E6   A        CLR    COUNT4
02577A CDAB 39                 RTS
02578A CDAC 86 01     A SPA99  LDAA   #01
02579A CDAE 97 E6     A        STAA   COUNT4
02580                          *
02581A CDB0 CE 0270   A        LDX    #00270
02582A CDB3 86 80     A        LDAA   #080
02583A CDB5 DF EA     A        STX    INDEX
02584A CDB7 97 E3     A        STAA   COUNT1
02585A CDB9 96 E3     A SPA17  LDAA   COUNT1
02586A CDBB 94 B2     A        ANDA   80A
02587A CDBD 26 67 CE26         BNE    SPA11
02588A CDBF E6 0A     A        LDAB   C2E,X
02589A CDC1 27 0A CDCD         BEQ    SPA49
02590A CDC3 6F 0A     A SPA59  CLR    C2E,X
02591A CDC5 6F 02     A        CLR    B2E,X
02592A CDC7 6F 8F     A        CLR    0CC+080,X
02593A CDC9 6F 03     A        CLR    B28,X
02594A CDCB 20 59 CE26         BRA    SPA11
02595A CDCD E6 02     A SPA49  LDAB   B2E,X
02596A CDCF 27 55 CE26         BEQ    SPA11
02597A CDD1 6F 02     A        CLR    B2E,X
02598A CDD3 D1 E8     A        CMPB   COUNT6
02599A CDD5 2B EC CDC3         BMI    SPA59
02600A CDD7 DF F4     A        STX    TEMP02
02601A CDD9 D7 F8     A        STAB   TEMP04
02602A CDDB A6 05     A        LDAA   LBA,X
02603A CDDD 97 F9     A        STAA   TEMP05
02604A CDDF BD C9E0   A        JSR    DIV
02605A CDE2 D6 FA     A        LDAB   TEMP06
02606A CDE4 D7 F5     A        STAB   TEMP01
02607A CDE6 DE F4     A        LDX    TEMP02
02608A CDE8 E6 03     A        LDAB   B28,X
02609A CDEA 6F 03     A        CLR    B28,X
02610A CDEC DF F4     A        STX    TEMP02
```

```
PAGE  046  LUCK    .BAIO  PCBBA

02611A CBEE B7 F8     A          STAB   TEMP04
02612A CBF0 A6 05     A          LDAA   LBA,X
02613A CBF2 97 F9     A          STAA   TEMP05
02614A CBF4 BD C9E0   A          JSR    DIV
02615                      B
02616A CBF7 86 01     A          LDAA   #01
02617A CBF9 97 EA     A          STAA   INDEX
02618A CBFB DE EA     A          LDX    INDEX
02619                      B
02620A CBFD B6 FA     A          LDAB   TEMP06
02621A CBFF BB F5     A          ADDB   TEMP01
02622A CE01 57                   ASRB
02623A CE02 B7 F5     A          STAB   TEMP01
02624A CE04 B7 F8     A          STAB   TEMP04
02625A CE06 A6 8F     A          LDAA   BCC+#80,X
02626A CE08 6F 8F     A          CLR    BCC+#80,X
02627A CE0A 97 F9     A          STAA   TEMP05
02628A CE0C BD C9B0   A          JSR    MULT
02629A CE0F B6 FB     A          LDAB   TEMP07
02630                      B
02631A CE11 86 02     A          LDAA   #02
02632A CE13 97 EA     A          STAA   INDEX
02633A CE15 DE EA     A          LDX    INDEX
02634                      B
02635A CE17 E0 07     A          SUBB   LBA,X
02636A CE19 B1 A3     A          CMPB   TRUCK
02637A CE1B 2B 17 CE34           BMI    SPA2
02638A CE1D 0C                   CLC
02639                      B
02640A CE1E 86 01     A          LDAA   #01
02641A CE20 97 EA     A          STAA   INDEX
02642A CE22 DE EA     A          LDX    INDEX
02643                      B
02644A CE24 20 04 CE2A           BRA    SPA9
02645A CE26 20 56 CE7E SPA11     BRA    SPA1
02646A CE28 20 8F CB89 SPA7      BRA    SPA17
02647A CE2A A6 6E     A SPA9     LDAA   LENL+#80,X
02648A CE2C 89 01     A          ADCA   #1
02649A CE2E A7 8E     A          STAA   LENL+#80,X
02650A CE30 24 02 CE34           BCC    SPA2
02651A CE32 6C 8B     A          INC    LENH+#80,X
02652                      B
02653A CE34 CE 0200   A SPA2     LDX    #0200
02654A CE37 B6 F5     A          LDAB   TEMP01
02655A CE39 C1 80     A SPA4     CMPB   SPD,X
02656A CE3B 2E 29 CE66           BGT    SPA12
02657A CE3D 2A 20 CE5F           BPL    SPA3
02658A CE3F 08             SPA6  INX
02659A CE40 8C 020B   A          CPX    #020B
02660A CE43 26 F4 CE39           BNE    SPA4
02661A CE45 0C                   CLC
02662A CE46 A6 C0     A          LDAA   SPED+#20,X
02663A CE48 89 01     A          ADCA   #1
02664A CE4A A7 C0     A          STAA   SPED+#20,X
02665A CE4C 24 0F CE5D           BCC    SPA5
02666A CE4E 0C                   CLC
02667A CE4F A6 B0     A          LDAA   SPED+#10,X
02668A CE51 89 01     A          ADCA   #1
```

```
PAGE  047  LUCK    .BAIO  PCBBA

02669A CE53 A7 B0     A          STAA   SPED+#10,X
02670A CE55 24 06 CE5D           BCC    SPA5
02671A CE57 A6 A0     A          LDAA   SPED,X
02672A CE59 8B 01     A          ADDA   #1
02673A CE5B A7 A0     A          STAA   SPED,X
02674A CE5D 20 1F CE7E SPA5      BRA    SPA1
02675                      B
02676A CE5F 96 FA     A SPA3     LDAA   TEMP06
02677A CE61 A1 90     A          CMPA   REM,X
02678A CE63 2A 01 CE66           BPL    SPA12
02679A CE65 08                   INX
02680A CE66 0C             SPA12 CLC
02681
02682A CE67 A6 C0     A          LDAA   SPED+#20,X
02683A CE69 89 01     A          ADCA   #1
02684A CE6B A7 C0     A          STAA   SPED+#20,X
02685A CE6D 24 0F CE7E           BCC    SPA1
02686A CE6F 0C                   CLC
02687A CE70 A6 B0     A          LDAA   SPED+#10,X
02688A CE72 89 01     A          ADCA   #1
02689A CE74 A7 B0     A          STAA   SPED+#10,X
02690A CE76 24 06 CE7E           BCC    SPA1
02691A CE78 A6 A0     A          LDAA   SPED,X
02692A CE7A 8B 01     A          ADDA   #1
02693A CE7C A7 A0     A          STAA   SPED,X
02694                      B
02695A CE7E 96 E3     A SPA1     LDAA   COUNT1
02696A CE80 47                   ASRA
02697A CE81 97 E3     A          STAA   COUNT1
02698A CE83 86 02     A          LDAA   #02
02699A CE85 97 EA     A          STAA   INDEX
02700A CE87 BD C030   A          JSR    NEXTLN
02701A CE8A 24 9C CE28           BCC    SPA7
02702A CE8C 39                   RTS
02703                            END
TOTAL ERRORS 00000--00000
```

```
C42A ABTBL   010918
C089 ABAIN   003280034B
00EB AR1HOL  000778
00EC AR1HOH  00074801275 01298 01325 01387 01410 01437
00EF AR1HVL  00079801332 01444
00EE AR1HVH  000788
C4E1 AVG     013178
D004 B1      00446800454 00457
0000 B1E     00110802002 02003 02433 02437 02441
0001 B1B     00111802009 02435 02450 02451
0002 B2E     00112802092 02093 02591 02595 02597
0003 B2B     00113802099 02593 02608 02609
D302 BEGIN   00793 00848800886
D01A C10     00461800470
D038 C11     00475 004778
D032 C12     00474800483
D020 C13     00462 004649
D132 C18     00626 006288
D137 C19     00629 006318
```

```
0008 C1E     00110002005 02430 02432
0009 C1B     001198
B114 C2      00608 006108
000A C2E     00120002095 02588 02590
000B C2B     001218
B119 C3      00611 006138
B11E C4      00614 006148
B123 C5      00617 006198
B128 C6      00620 006228
B12D C7      00623 006258
B0F9 C87     00592800601
B0E1 C88     00581800589
B14A C9      00641 006448
C084 CHECK   00338 00342 00344 003478
C000 CNP14   00152800219 01544
00E3 COUNT1  00069802424 02427 02538 02540 02584 02585 02695 02697
00E4 COUNT2  000708
00E5 COUNT3  00071802416 02418 02421
00E6 COUNT4  00072800497 02574 02576 02579
00E7 COUNT5  000738
00E8 COUNT6  00074802415 02439 02573 02598
009F CYCLE   00043800500 00781 00783 00787 00791 00849 00876 00878 00882 00884
C409 D1      01068801074
C40B D2      01069801072
C438 D3      01096 010988
C441 D5      01101 011038
C42B D50     01092801122
C453 D56     01111 011138
C849 DA1     01182801188
C84D DA2     01183801184
C87A DA3     01210 012128
C883 DA5     01215 012178
C86F DA50    01206801236
C895 DA56    01225 012278
0005 DAOFNO  00004800700 00710 00720 00894 00904 00914
0009 DAOFWK  00005800702 00712 00722 00896 00906 00916
00BC DATAOL  00063801928
00AC DATAOM  00053800955 00956 00969 00971 01806 01807 01820 01822 01930 01933
00BD DATAIL  000648
00AD DATAIN  000548
007F DAYMAX  00035800518
000A DET1    00103801283 01288 01395 01400 01501 01516 01536 01537 01591 01604 01626 01627 02034 02035
             02119 02120
0088 DETER1  00020801519 01520
0089 DETER2  000218
008A DETER3  00022801609 01610
008B DETER4  000238
00BE DETINL  00065802104 02220
00AE DETINH  00055802014 02164
C9E0 DIV     00170800403 02446 02456 02604 02614
CA3E DO      01833 018388
C2BE DONE    00982 009878
00D0 DTCTNL  00080800492 00494
00C0 DTCTNH  00067800488 00490 00970 01821 01933
CA3F DTRTB   01808 018438
C2BF DTRTBL  00957 009928
C4B5 E3      01284801292
C4BD E5      01285 012918
```



```
C8E5 EA3     01378801404
C8ED EA5     01397 014038
C5A1 ER9     01535 015388
C9A4 ER90    01625 016288
0080 ERRBIT  01487801508 01598
C599 ERSET   01500 01509 015348
C99C ERSETT  01590 01599 016248
C6B0 G1      01660 016448
C4FE G10     01710 017168
C6D1 G11     01687801690
C6F6 G15     01707 017118
C702 G20     01718801754
C740 G30     01723 017548
C744 G35     017588
C74B G36     01759 017628
C736 G40     01724 01747 017508
C70E G45     01722 017268
C6B6 G5      01662 01667 016708
C6E2 G50     01699801766
C6CA G51     01680 016828
C726 G52     01736 017388
C754 G60     01712 017688
C6BB G7      01674801683
C6A6 G70     01656 016588
C6A8 G71     01657 016598
C6CD G9      01675 016858
CF10 GA1     02277 022838
CF5E GA10    02327 023338
CF31 GA11    02304802307
CF56 GA15    02324 023288
CF62 GA20    02335802372
CFA0 GA30    02340 023748
CFA4 GA35    023768
CFAB GA36    02377 023808
CF96 GA40    02341 02365 023688
CF4E GA45    02339 023438
CF16 GA5     02279 02284 022878
CF42 GA50    02316802384
CF2A GA51    02297 022998
CF86 GA52    02354 023568
CFB4 GA60    02329 023862
CF1B GA7     02291802300
CF06 GA70    02273 022758
CF08 GA71    02274 022768
CF2D GA9     02292 023028
00B0 GO      00058801998 02029 02030 02174 02175 02428
00B2 GOA     00060802088 02114 02115 02230 02231 02584
00B1 GOT     00059802007 02044 02045 02183 02184
00B3 GOTA    00061802097 02128 02129 02239 02240
B2B3 H1      00799 008018
B336 H10     00881 008948
B2B9 H11     00820 008228
B2C6 H2      00809 008118
B2EC H20     00830 008328
B2A7 H3      00795800887
B28B H30     00773 007798
B259 H31     00752800777
B33F H4      00847 00874 008898
```

```
PAGE  050  LUCK    .BAIO  PCBBA

   D2A1 HP      00786 00791#
   D250 HEADER  00158#00746
   000A HR      00004#01455 02272
   00EA INDEX   00075#00242 00245 00247 00262 00950 00951 00968 00981 00984 01126 01134 01327 01329 01439
                01441 01540 01545 01801 01802 01819 01832 01835 02425 02459 02460 02475 02476 02484 02485
                02542 02383 02617 02618 02632 02633 02641 02642 02699
   D000 INIT    00157#00434
   C2A0 INPUT   00159#00945 01912
   CA00 INPUTT  00171#01796 01913
   CA12 IP1     01810#01836
   CA14 IP2     01812#01831
   CA38 IP3     01834#
   CA33 IP5     01832#
   C2B2 IPT1    00959#00985
   C2B4 IPT2    00961#00980
   C2B8 IPT3    00983#
   C2B3 IPT5    00981#
   00F4 KAY1    00004#00501 01065 01123
   00A2 KAY2    00047#00502 01179 01237
   0006 LB      00116#00593 02478
   0007 LDA     00117#00594 02635
   0012 LDESW1  00010#00489
   0013 LDESW2  00011#00493
   001E LDHW1   00015#00461 00487
   001F LDHW2   00016#00474 00491
   007A LENDAY  00032#00499 00790
   00A4 LENGTH  00049#00458 00662 00676 00858
   000E LENL    00107#02344 02346 02490 02492 02647 02649
   000B LENM    00106#02343 02345 02494 02651
   007D LNBUF1  00034#00665
   007B LNDAY1  00033#00678
   000C LNDIS1  00105#00463 00476 01291 01403 01534 01424
   C55E LO1     01497#01511
   C585 LO11    01518 01521#
   C57C LO12    01516#01530
   C598 LO13    01524 01531#
   C566 LO7     01498 01501#
   C570 LO9     01507 01510#
   C95E LOA1    01587#01601
   C980 LOA11   01608 01611#
   C97F LOA12   01606#01620
   C99B LOA13   01614 01621#
   C968 LOA7    01588 01591#
   C973 LOA9    01597 01600#
   C4A5 LOOP1   01276#01279
   C4B0 LOOP2   01283#01315
   C4CA LOOP3   01299#
   C4E5 LOOP4   01319#01322
   C4F0 LOOP5   01326#01340
   C8B5 LOOPA1  01388#01391
   C8E0 LOOPA2  01395#01427
   C8FA LOOPA3  01411#
   C915 LOOPA4  01431#01434
   C920 LOOPA5  01438#01452
   0004 LS      00114#00581 02444 02454
   0005 LSA     00115#00582 02602 02612
   C03E LSTLN   00243 00250#
   C9E4 MD1     00408#00415
```

```
C9EF MD2     00411 00414#
C9B5 MM1     00386#00393
C9C0 MM2     00389 00392#
0004 MONTH   00003#00498 00708 00718 00892 00902 00912
006E MRAM    00018#
C9B0 MULT    00169#00381 00651 00656 02471 02628
C030 MEXTLN  00153#00240 01073 01088 01118 01187 01202 01232 01314 01426 01510 01600 02049 02133 02188
             02244 02543 02700
0010 MLANE1  00008#
0011 MLANE2  00009#
C4BC MLOOP2  01289 01310 01314#
C90C MOOP2   01401 01422 01424#
0014 MVOL1   00012#01492 02309
0015 MVOL2   00013#01711 02328
CA29 MXDET   01818 01824#
C2C9 MXTDET  00967 00973#
0082 O1DIT   01484#01499 01589
009C OC10CL  00040#
009B OC10CM  00039#00511 01494 01584
000F OCC     00108#02040 02125 02434 02468 02469 02592 02625 02626
0006 OCCIL   00099#01094 01095 01208 01209
0005 OCCIM   00098#01097 01211
0003 OCMM1L  00096#01099 01100 01213 01214 01297 01409
0002 OCMM1M  00095#01079 01102 01193 01216 01295 01407 01489 01579
C5A2 OCTEST  01497 01540#01587
C557 OCTST1  01494#
C012 ONEBIG  00224 00228 00232#
C00F ONESML  00225 00229#
0000 OSIX1   00093#01069 01092 01103 01183 01206 01217 02038 02123
C506 OUT     01333 01342#
C507 OUT1    01343#
C934 OUTT    01445 01454#
C937 OUTT1   01455#
C0C3 P1      00334 00337#
C0B5 P10     00310 00316#
0000 PCLN1   00130#01345
0080 PCLN2   00131#01346
00AB PCLN3   00132#01347
00C0 PCLN4   00133#01348
00CB PCLN5   00134#01349
00D5 PCLN6   00135#01350
00DB PCLN7   00136#01351
00E0 PCLN8   00137#01352
C350 PCS2    00140#01911
D4A0 PCS3    00141#01989
D550 PCS3A   00172#02078
CBB0 PCS3B   00173#02140
CC00 PCS3C   00174#02215
C400 PCS4    00142#01061
CB40 PCS4A   00144#01175
C4A0 PCS5    00143#01270
CBD0 PCS5A   00147#01382
C550 PCS6    00144#01484
C950 PCS6A   00148#01574
C6A0 PCS7    00145#01653
CF00 PCS7A   00177#02370
CC50 PCS8    00175#02412
CDA0 PCS8A   00176#02570
```

```
0000 PCSCAB 00001#
00F2 PCW1   00083#00504 01078 01081 01192 01195
C506 PCWTAB 01317 01344#01429
C91B PICKUP 01432 01436#
C4EB PIKUP  01320 01324#
C859 POP1   01180 01192#01203
5001 PORT1  00181#
5002 PORT2  00182#
5003 PORT3  00183#
5004 PORT4  00184#
C472 PRO14  01080 01134#01194
C444 PROB   01083 01126#01197
C419 PROP1  01066 01078#01089
C0B0 PRPRT  00156#00312 01137 01328 01440
C080 PRPRTB 00155#00305 01129
009B PTR    00041#00674 00805 00808 00810 00811 00814 00856 01670
0096 PTRL   00037#00683 00826 00829 00831 00832 00835 00864 02287
0098 PTRS   00038#00691 00754 00756 00761 00763 00768 00775 00845 00872
0090 REN    00123#02520 02677
B045 RET    00487#
C35E S0     01914 01924#
C36B S1     01932#01953 01940
C060 SETUP  00154#00240 01063 01076 01091 01177 01190 01205 01282 01394 01496 01586 01990 02080 02141
            02217
0095 SIZE   00036#00632 01676 01686 02293 02303
B327 SMALL  00876#
CB2E SP1    02488 02517 02528 02533 02538#
CCB6 SP11   02429 02436 02438 02488#
CB16 SP12   02499 02521 02524#
CC69 SP17   02427#02489
CCE4 SP2    02480 02493 02496#
CB0F SP3    02500 02519#
CCE9 SP4    02498#02503
CC7D SP49   02431 02437#
CB0D SP5    02508 02513 02517#
CC73 SP59   02432#02440
CCEF SP6    02501#
CCD8 SP7    02489#02544
CCDA SP9    02487 02490#
CC5C SP99   02417 02420#
CE7E SPA1   02645 02674 02685 02690 02695#
CE26 SPA11  02587 02594 02596 02645#
CE66 SPA12  02636 02678 02680#
CDB9 SPA17  02585#02646
CE34 SPA2   02637 02650 02653#
CE5F SPA3   02657 02676#
CE39 SPA4   02655#02660
CDCD SPA49  02589 02595#
CE5D SPA5   02665 02670 02674#
CDC3 SPA59  02590#02599
CE3F SPA6   02658#
CE28 SPA7   02646#02701
CE2A SPA9   02644 02647#
CDAC SPA99  02575 02578#
0078 SPBUF  00031#00509
00A0 SPC1   00045#01705 01717 01753 02322 02334 02371
00A1 SPC2   00046#01709 01714 01750 01752 02326 02331 02368 02370
00B0 SPD    00122#00523 00524 00528 00529 00533 00534 00538 00539 00544 00545 00549 00550 00554 00555
```

```
PAGE  053  LUCK    .SA10  PC88A

                00559 00560 00564 00565 00570 00571 00575 00576 02498 02655
0076 SPDAY1     00030800507
0075 SPDDAY     00029800498 00789
00A0 SPED       00124800752 00753 00759 00760 00766 00767 02505 02507 02510 02512 02514 02516 02525 02527
                02530 02532 02534 02536 02662 02664 02667 02669 02671 02673 02682 02684 02687 02689 02691
                02693
00A6 SPEED      00050800513 00685 00866
0409 START      00124800666 00850
0009 STCHG1     00104802032 02117
00BF STCHGL     00064802109 02126 02225 02235
00AF STCHGM     00054802019 02026 02027 02041 02169 02179
04A7 T1         01994802050
D4F8 T14        02020 02036 02042 020478
D4EC T16        02015 020388
D4D2 T2         020198
D4BF T20        01999 02004 020068
D4C7 T21        02008 020118
D5B7 TA1        02084802134
D5A2 TA14       02110 02121 02127 021318
D594 TA16       02105 021238
D5B2 TA2        021098
D56F TA20       02089 02094 020948
D577 TA21       02098 021018
CDD5 TB1        02164802189
CDD1 TB14       02170 02177 02180 021878
CDC6 TB16       02165 021798
CDD9 TB2        021698
0001 TBIT0      00141800993 01010 01844 01861
0002 TBIT1      00142800995 01012 01846 01863
0004 TBIT2      00143800997 01014 01848 01865
0008 TBIT3      00144800999 01016 01850 01867
0010 TBIT4      00145801001 01018 01852 01869
0020 TBIT5      00146801003 01020 01854 01871
0040 TBIT6      00147801005 01022 01856 01873
0080 TBIT7      00148801007 01024 01858 01875 01991 02081 02162 02218
CC05 TC1        02220802245
CC21 TC14       02226 02233 02234 022438
CC16 TC16       02221 022358
CC09 TC2        022258
00F5 TEMP01     00085800451 00452 00464 00468 00477 00481 00583 00587 00595 00599 00606 00637 00638 00640
                00948 00965 01525 01529 01615 01619 01693 01706 01758 01761 01799 01816 01994 01995 01996
                02011 02012 02013 02084 02085 02086 02101 02102 02103 02310 02323 02376 02379 02448 02464
                02466 02497 02606 02621 02623 02654
00F6 TEMP02     00086800223 00455 00466 00467 00479 00480 00584 00585 00596 00597 00763 01495 01526 01528
                01585 01616 01618 01703 01716 01718 01721 01814 02320 02333 02335 02338 02442 02449 02452
                02600 02607 02610
00F7 TEMP03     00087800227 00653 00659 00749 00771 00772 01515 01521 01523 01605 01611 01613 01695 01731
                01732 01757 01992 01997 02006 02048 02082 02087 02096 02132 02312 02348 02349 02375
00F8 TEMP04     00088800388 00410 00412 00448 00655 00958 00973 00976 01064 01128 01130 01136 01138 01178
                01490 01541 01580 01671 01681 01685 01737 01739 01809 01824 01827 01925 01952 01958 02288
                02298 02302 02355 02357 02443 02453 02467 02601 02611 02624
00F9 TEMP05     00089800390 00407 00650 01127 01135 01543 01678 01679 01734 01735 01927 01959 02295 02296
                02351 02352 02445 02455 02470 02603 02613 02627
00FA TEMP06     00090800394 00416 00660 00751 00758 00765 00776 01542 01696 01699 01929 01951 01957 02313
                02316 02447 02463 02519 02605 02620 02674
00FB TEMP07     00091800395 00417 00652 00657 00774 01697 01762 01764 01954 01956 02314 02380 02382 02472
                02629
00A3 TRUCK      00048800516 02479 02636
```

```
0009 VDIL   00102801109 01110 01223 01224 01727 01729
0008 VDIN   00101801112 01224 01726 01728
0004 VNN1   00097801082 01115 01116 01196 01229 01230 01293 01405
0073 VOBUF1 00028800505 00643 00645 00647 00654 00664 00796 00802 00806 00812 00817 00823 00827 00833
0091 VODAY1 00027800667 00695 00795 00798 00800 00801 00804 00851 00889
00AB VODAYL 00051800677 00705 00816 00819 00821 00822 00825 00859 00899
00AA VODAYB 00052800686 00715 00837 00839 00867 00909
0007 VOL1   00100801104 01107 01220 01221
0070 VOLDAY 00025800496 00784 00788 00879 00883
0016 VOLDW1 00014800607 00610 00613 00616 00619 00622 00625 00628 01701 02318
0001 VSIX1  00094801104 01117 01218 01231 02022 02112
0071 WW     00026800468
00F2 XLANE  000826
0003 YEAR   00002800696 00706 00716 00890 00900 00910
```